ELECTRONIC COMPUTERS

Principles and Applications

T. E. Ivall

A 'WIRELESS WORLD' BOOK

ABOUT THIS BOOK

This is the second edition of a book first published in 1956 and so well received that it had to be reprinted within twelve months. Developments since then have been so rapid that the present volume has had to be almost entirely rewritten, most of the existing diagrams replaced by new ones, and pictures of more modern electronic computer equipment introduced. Three new chapters have also been added, dealing respectively with analogue computer circuits, the programming of digital computers, and the evolution of the more " intelligent " machines of the future. Advantage has also been taken to improve the exposition of basic principles in the earlier chapters.

The fundamental approach to the subject has, however, remained unchanged and as before the book is intended as a non-mathematical introduction to the principles and applications of computers employing valves, transistors and other electronic devices, and has been designed to appeal to technicians, engineers and students who have some knowledge of electronic or electrical engineering; but some chapters are also suitable for the interested layman.

The treatment has been deliberately made as general as possible in order to give a broad background picture of the whole field of computing.

The bulk of the book is devoted to describing the circuitry and construction of both digital and analogue computers, but their rapidly developing applications in industry, commerce and science are also outlined. Here considerable emphasis is placed on the application to " automation," or " control," techniques in industry, and also on the computing techniques which are nowadays playing an important part in research and design work.

The author is well known to a wide circle of friends as joint-Assistant Editor of " Wireless World."

260 *pages, including* 32 *pages of art plates.*

25s. net

ELECTRONIC COMPUTERS

PRINCIPLES AND APPLICATIONS

ELECTRONIC COMPUTERS
PRINCIPLES AND APPLICATIONS

T. E. IVALL

LONDON : ILIFFE & SONS LTD
NEW YORK : PHILOSOPHICAL LIBRARY

First published 1956
Second impression 1957
***Second Edition* © *Iliffe & Sons, Ltd.*, 1960**

**Published for " Wireless World " by Iliffe & Sons, Ltd.,
Dorset House, Stamford Street, London, S.E.1.**

**Published in the U.S.A. by Philosophical Library Inc.,
15 East 40th Street, New York 16, N.Y.**

**Made and Printed in England at The Chapel River Press,
Andover, Hants.**

BKS/3435

CONTENTS

Preface vii

Chapter 1
Evolution of the Computer 1

Chapter 2
General Principles of Computing 14

Chapter 3
Analogue Computing Circuits—1 34

Chapter 4
Analogue Computing Circuits—2 45

Chapter 5
Equipment of Analogue Computers 57

Chapter 6
Applications of Analogue Computers 75

Chapter 7
Digital Computer Circuits—1 97

Chapter 8
Digital Computer Circuits—2 114

Chapter 9
Storage Systems 132

Chapter 10
Digital Computer Equipment 156

Chapter 11
Programming Digital Computers 183

Chapter 12
Applications of Digital Computers 197

Chapter 13
Recent Developments 217

Chapter 14
Computers of the Future 245

Index 261

PREFACE

THE general trend towards automation in industry and commerce has created a good deal of interest in electronic computers, which, through their spectacular achievements, their speed and complexity of operation, have become popularly known as " electronic brains ". We recognize the electronic computer as being at the very core of automation, even though we may not be very clear what automation really means. It is the computer which makes possible the automatization of brain-work, and so adds something entirely new to the gradual processes of mechanization which have been going on since the Industrial Revolution and before. This is particularly true in the sphere of office work, where the impact of the computer on the established methods of mechanization has been largely responsible for the emergence of the new subject called data processing.

This book is an introduction for those who are beginning to take an interest in electronic computers. It is not, therefore, a book for computer experts. Nor is it a text-book. Some effort has been made to write in fairly general terms, but with a subject in a state of such rapid development it is impossible to lay down the law about standard practice. At the same time, of course, there are certain principles which do not change.

What the book *does* try to do is to give a reasonably broad picture of electronic computing to those who are likely to become involved in some specialized aspect of it, either as users or as electronic designers, in the immediate future. For this reason the author makes no apology for including analogue computing and digital computing, circuit techniques and practical applications, all within the same two covers. A grounding of electronic or radio techniques is assumed in the reader, but non-technical people will probably be able to manage Chapters 1, 2, 6, 12 and 14, which form almost a connected story without all the details of electronic design.

This second edition has been largely re-written, partly to provide a more coherent picture than was possible with the separate contributions of the first edition and partly to bring the book up to date on recent developments. Three new chapters have been added, one on analogue computing circuits, one on digital computer programming and one on recent technical developments. Although very little of the original contributors' work remains, their inspiration is still behind the book and the present author is most grateful for that. One contribution has, however, been kept almost intact—the explanation of general principles by " Cathode Ray " which forms the basis of Chapter 2. It would be presumptuous to attempt to improve on the expository style of " Cathode Ray " or even to try to imitate it.

Finally the author would like to thank all those firms who have kindly provided technical details and photographs of their products.

T. E. IVALL.

London, S.E.1.

I

EVOLUTION OF THE COMPUTER

NOT many years ago the electronic computing machine was little more than a highbrow laboratory instrument, to be found only in the more advanced mathematics departments of universities or government research establishments. Constructed in amateur fashion from war " surplus " components, it was generally in a state of modification or half completion, and was seldom accompanied by anything in the nature of a full circuit diagram. All that could be found were schematics using strange-looking symbols which no doubt could be understood by computer people but certainly not by anybody else.

Apart from everything else, the habit of referring to the machine as a " calculating engine ", metaphorically placing it alongside Stephenson's " Rocket " and other wonders of the Industrial Revolution, made it seem even more remote from the everyday practical world.

Nowadays you can buy electronic computers " over the counter "—anything from a small desk-type machine worth a few hundred pounds to a roomful of equipment costing tens of thousands that will perform nearly all the functions of a large commercial accounts department. These machines are constructed on the best engineering principles and come in sleek, grey cases with instruction manuals covering every detail, while technical salesmen explain to the customer in practical terms exactly how computing apparatus can be applied to his particular problems.

Some of the machines are being produced by well-established electronics firms and instrument manufacturers, and at the same time new companies are being formed whose sole business is to make and sell computers. An interesting example is Leo Computers, Ltd., which was formed by the Lyons' teashop organization after the successful application of a large-scale computer (known as LEO, or Lyons' Electronic Office) to their head-office accounting work. Computers, in fact, are becoming big business, and it is quite common to find them mentioned nowadays in the financial columns of the newspapers.

But what is the reason for this sudden popularity? Hard-headed business men do not usually sink their money into such ventures unless there is a definite demand. The answer is, of course, that a new and advanced form of mechanization is beginning to appear in industry and commerce—mechanization of brain-work. Whereas in the Industrial Revolution it was man's muscles that were replaced by mechanical power, in this latest phase, which some call the Second Industrial Revolution, we see machines taking over some of his simpler mental processes. Whether such machines can be said to " think " depends on how you define the word. But there is no doubt about the fact that they will perform with great speed and accuracy the sort of routine logical operations which constitute a large part of the brain-work of semi-skilled workers in factories and offices nowadays. More and more people are being used for this type of work, but with the present shortage and cost of labour, managements are finding it increasingly uneconomic to employ human beings when there are machines which can do the same operations faster, cheaper and more reliably.

Here, then, is the main demand for electronic computers. At the moment it comes principally from large office organizations, since a great deal of clerical work consists of putting numerical information through various logical processes (calculation being just one of them) and these are particularly well adapted to mechanization in a computer. Moreover, clerical workers have been accustomed to the idea of calculating machines—of the mechanical kind—for a good many years now, and the high-speed electronic computer comes to them more as a natural development than an innovation.

From the manufacturing side of industry the demand for computing machinery is very small at the moment. A variety of computer-like devices and circuits are used in electronic control mechanisms, but these perform very specialized operations and bear little resemblance to the comparatively large and versatile office machines. There is, however, considerable enthusiasm in the mass-production industries for anything connected with " automation " and " the automatic factory ", and a characteristic feature of this philosophy is the use of computers for the direct control of industrial plants. This is a thing of the future, of course, but a start has already been made with computer-controlled machine tools, in which the computer automatically works out in terms of mathematical functions the contours to be machined and so provides information to control the cutting tool. Moreover in some factories computers are being used to assemble and " process " the data on which the control of the plant depends.

Of course there has always been a demand for computers in research establishments. Here one finds probably the widest range of applications, for practically all branches of scientific research require calculations to be done at some time or other—often of the most tedious kind. In fact, quite a few scientific and technical problems have remained unsolved for years simply because the calculations were too formidable for a human being to tackle without losing his sanity, and it is only now, with the use of computers, that the answers are being found. Aircraft design, ballistics, crystallography, electron optics, astronomy, pure mathematics and weather forecasting are just a few of the activities in which computing machines are playing important roles. And the scope will undoubtedly widen in the future. The actual number of machines used in research work, however, is never likely to be large, and one can see that the main impetus to the development of new techniques will come from industry and commerce.

Calculating machines as such are by no means new, of course. If one leaves out the abacus or counting frame, which appeared long before Christianity, they are about 300 years old or more. John Napier, a Scot, must be given credit for paving the way, at the beginning of the seventeenth century. He not only devised a system of numbering rods as an aid to multiplication (known as Napier's " bones ") but was the inventor of logarithms, which were almost immediately embodied in the slide-rule. The significance of this double achievement was that the " bones " were an elementary aid to digital calculation while the slide-rule was probably the earliest analogue device, and these represent the two main streams in computer technology as we know it today. In an analogue computer each number is represented by the size of a physical quantity, say voltage, whereas in a digital machine it is coded as a succession of signals, rather like the pulses from an automatic telephone dial. More will be said in the next chapter about the distinction between the two types.

Napier's " bones ", however, had no real practical value, and were little more than a scientific curiosity. It was a Frenchman, Blaise Pascal, who achieved the first successful mechanization in 1642. His machine consisted of a series of wheels with numbers round them, the first wheel representing units, the second tens, the third hundreds, and so on, and it worked on the same sort of principle as the modern revolution counter. Although it was a very simple device it contained one important development which is an essential part of digital computing technique—a means of making the " carry over ", used in adding numbers together, into an automatic process. Moreover, it showed the possibility of multiplying numbers by successive addition—another principle used in modern computers—and this was actually embodied in a machine by Leibnitz in 1671. From then onwards a variety of improved calculating machines were devised, all based on the original idea of wheels moved round in steps, and in the nineteenth century commercial models began to appear.

Burkhardt in Germany, Odhner in Sweden and Burroughs in America were three of the important developers, and the results of their work are embodied in mechanical calculators of the kind we see today in banks, offices, shops and laboratories all over the world.

These machines, however, cannot really be considered as true precursors of the modern electronic digital computer. They are limited in scope and require the constant attention of a human operator to supply them with numbers, to work the machinery and to record on paper the result of each stage of the calculation before proceeding to the next one. Although they make possible great savings in time and mental effort and are admirably suited to certain types of work, they are still restricted in speed by the capabilities of the human operator in many respects. And this question of speed becomes really important when large quantities of numerical data have to be dealt with.

It was Charles Babbage, the nineteenth-century Cambridge mathematician, who really laid down the principles of our modern computers in mechanical form. A significant part of his design for an " analytical engine " was the use of a " store " for holding the partial results of arithmetic operations performed in what he called the " mill ". To feed numbers into the machine he proposed a system using patterns of holes punched in cards, while the results of calculations were to be presented by a mechanism which would set up the appropriate numerals in type ready for printing. Another proposal was that the " engine " should be capable of " deciding " on a course of action in the light of partial results obtained.

All these ideas have become features of the modern digital computer, and it is mainly because of them that our present-day machines are so fast in operation and so versatile in their range of calculations. Unfortunately Babbage never completed his " engine ", and it is doubtful whether it would have actually worked if he had, owing to the limitations of the mechanical engineering of those days.* In this respect the inventor was well ahead of his time, and it was only with the coming of electrical and electronic techniques that the necessary logical operations could be achieved without immense practical difficulties.

At about the same time as Babbage was working on his machine another mathematician was putting forward some new ideas which have since found expression in the modern counterparts of the " analytical engine ". This was George Boole, and he was pioneering the science of symbolic logic—that is, using symbols to represent ideas and manipulating them in a kind of algebra to produce answers to logical problems. The processes of symbolic logic are, in fact, very much akin to the operations of a modern computer—though the present age cannot really claim the credit for this mechanization. In 1866 a certain W. S. Jevons conceived the idea of a " logical abacus " using wooden tablets, each inscribed with the symbols for a particular statement or proposition, which could be moved about in a frame. He added a keyboard and apparatus for manipulating the tablets and so produced a complete machine for solving logical problems, which was demonstrated to the Royal Society in 1870. This mechanization of logic is not really in the same line of development as the mechanization of numerical calculation, which is more a matter of counting and arithmetic. It is interesting, however, because the modern digital computer can be used in much the same way as the Jevons machine.

---

* His son, H. P. Babbage, did, however, construct a part of the " mill " or arithmetic unit. This, and a few bits of Babbage's original mechanism, can be seen in the Science Museum at South Kensington, London.

Another significant development sprang from the punched cards with which Babbage proposed to feed numbers into his machine. These were actually based on cards invented by a Frenchman, Jacquard, for controlling complex loom operations. In 1889 the idea was taken up by Hollerith in America and was used in a sorting device which now forms the basis of most of the punched-card machines to be seen in modern business offices. In one type of machine the cards are " read " and calculations performed by mechanical methods, while in another kind everything is done by electromagnetic relays.

The development of analogue computers during this time, though unspectacular by comparison, followed much the same pattern—from mechanical aids to calculation to devices capable of solving complex problems, with electrical techniques providing greater versatility later on. If we leave out a mechanical problem solver attributed to Plato, the first really notable analogue device was the slide-rule, which, as already mentioned, was a mechanization of Napier's logarithms. In the first place the logarithms were plotted on scales by a certain Gunter, who performed multiplication and division by the addition and subtraction of lengths using dividers. It was Robert Bissaker, however, who made the first genuine slide-rule in 1654. From then until the electrical era mechanical devices were invented in great variety for different purposes. Systems based on the balancing of weights or of water levels in connected vessels were used for solving equations during the nineteenth century. The integraph was devised for working out differential equations, the planimeter for calculating areas under curves and other machines for such operations as differentiation and harmonic analysis.

Lord Kelvin invented a most elaborate device for predicting tides, with parts and movements representing the action of the sun and moon, the whole thing being driven by an electric motor. From the same hand came a machine for solving any number of linear simultaneous equations and also a mechanical integrating device. Felix Lucas goes on record as having worked out equations by electrostatic and electromagnetic methods in 1888, while Arthur Wright in 1909 devised a system for adding and subtracting based on the laws by which resistances combine in series and parallel. An analogue device of considerable importance was the differential analyser (a machine for solving differential equations), which was developed by V. Bush in America in the nineteen-twenties, perhaps from some original theoretical work by Lord Kelvin about 50 years before. Also in the nineteen-twenties, electrical engineers began to make use of resistance networks, now known as " network analysers ", for studying the problems of power distribution.

One could go on citing examples of this kind right up to the present day. The really big surge of development in analogue computing, however, came during the second world war, when techniques in great variety were devised for doing the ballistics calculations necessary for aiming guns, bombs and other projectiles. Such is the pace of modern warfare that these calculations have to be done at extremely high speed, and the analogue computer is ideal for this purpose, for it generally produces the results of its calculations instantaneously. The familiar " predictor " to be seen at anti-aircraft posts during the war was, in fact, a mechanical analogue computer.

During this time the rapid development of radar produced many new electronic circuits based on thermionic valves, and a whole body of technique was built up which provided much new material for the computer scientists, amongst others, to draw upon. All-electronic predictors began to appear, using electrical voltages instead of mechanical displacements to represent the variables. Electronic valve amplifiers for handling these direct voltages were greatly improved, and within a few years of the end of the war the techniques developed for specialized military purposes had broadened out into electronic analogue computing in the most general sense.

Nowadays one the biggest fields of application for the analogue computer is in the design of aircraft, and it is no accident that some of the most successful machines have been produced by the electronics departments of aircraft manufacturers. A particularly valuable function of the computer in this type of work (as in other fields) is to act as a " simulator " to give results which could otherwise only be obtained by staging elaborate and expensive tests on actual aircraft. Plate 2 (p. 18) shows a typical modern analogue machine.

Returning to the story of the digital computer, after Babbage's ill-starred efforts to produce a universal calculating machine (one that would cope with a variety of different problems) there were no more attempts in this direction for about 70 years. Meanwhile, however, there was considerable development in calculating machines for specialized purposes—notably commercial accounting—and names like Burroughs, Hollerith, Powers-Samas and Remington-Rand became well known in business offices. Although the machines were often extremely complex in operation and offered a great many useful facilities, they were still all based on the fundamental idea of Pascal's ten-position wheels. None of the manufacturers attempted to produce anything in the nature of a " mathematical " computer in the early part of this century, although in fact one or two of the machines proved capable of doing scientific and statistical work.

In 1937, however, the International Business Machines Corporation in America collaborated with Howard Aiken, of Harvard University, in the construction of an " automatic sequence controlled calculator ". This arose from Aiken's idea that the techniques and components used in business machines could be adapted to the construction of an automatic type of calculator controlled by a sequence of instructions. As in Babbage's " engine ", ten-position wheels formed the basis of the design, but the drive problem was solved in a much more sophisticated way by connecting them to rotating shafts as required by magnetically-operated clutches. The sequence control of the arithmetical operations was achieved by information fed in on a punched tape.

About the same time the Bell Telephone Laboratories began to explore the possibilities of electromagnetic relays for computing, and since then a number of machines have been built on this principle. It was not until the time of the second world war, however, that the advantages of the much faster electronic relay, using valves, were realized—even though the relay circuit itself (now known as a " trigger " circuit) had been invented by Eccles and Jordan as far back as 1919.

The first digital computer to make use of valves was the " Electronic Numerical Integrator and Calculator ", or ENIAC, designed by Eckert and Mauchley in America. Altogether it used about 18,000 valves and consumed vast quantities of electricity, but as a result of the new electronic techniques it could work at incredible speeds, doing additions and subtractions at the rate of about 5,000 a second.

In the ENIAC the ten-position wheels of the earlier mechanical calculators were transformed into rings of ten electronic relay circuits—rings, that is, in the electrical sense, not the mechanical. Each relay circuit was a two-state " trigger " device as invented by Eccles and Jordan: it could be either " on " or " off " but was normally in the " off " state. The equivalent process of a counter wheel rotating in the mechanical calculator was the trigger circuits' switching " on " in succession—the first triggering the second, the second triggering the third and eventually the tenth triggering the first again to complete the electrical rotation. Twenty storage registers based on this principle were used and each held a ten-digit decimal number. The interesting point was, however, that although the ENIAC worked in decimal numbers, it used two-state or *binary* circuits to represent them—thereby forming a link with the modern digital computers, which also use two-state circuits but represent the numbers in binary notation.

It is easy to see, then, why the ENIAC needed such a large number of valves, since a ring of ten electronic circuits, each comprising two valves, was required for every digit stored. This, of course, was an extremely cumbersome method (though useful for some purposes) and it was soon realized that there were much easier ways of storing information.

One idea was to keep the information continually circulating in a loop consisting of an acoustic delay-line with its output connected back to its input. This first came to fruition in 1949 in a machine called the " Electronic Delayed Storage Automatic Computer ", or EDSAC, which was built at Cambridge University under the direction of M. V. Wilkes. The reduction in the number of electronic relays by the new method of storage was considerable, and the EDSAC had only about 3,000 valves. Another machine using the same principle was completed shortly afterwards—the small " pilot model " of the Automatic Calculating Engine, or ACE, built by the National Physical Laboratory at Teddington. The input and control equipment of this early model is shown on p. 17 (Plate 1[a]), while the final ACE, which was completed in 1958 and still uses the delay-line storage principle, is on p. 19 (Plate 3[b]).

A significant feature of these machines—and of the American EDVAC (Electronic Discrete Variable Automatic Computer) completed at about the same time as the EDSAC—was that the instructions for controlling the sequence of operations were represented by a code of numbers.

This technique has now become standard practice in the larger modern digital computers. Another great saving of electronic valves was achieved by the introduction of binary notation to represent the numbers, since only one two-state circuit is required to store a binary digit.

During the same period other large "mathematical" machines were being developed, and there are now so many that it is impossible to keep track of them. Commercial firms soon began to see the possibilities in computing, and they started by producing engineered versions of some of the existing machines designed by universities and government establishments. Now they are designing new machines for themselves (see p. 17, Plate 1[b]). Meanwhile the business machine manufacturers had become interested in electronics and had started to produce fairly small and specialized calculators such as electronic multipliers. This gave them an insight into the practical possibilities of automatic "stored programme" computers, and they began to make machines of this advanced kind which were based on the "academic" designs but modified to fit into existing punched-card office installations.

In Britain, however, the business machine manufacturers did not then have quite the same background experience and facilities for electronic development work as the electronics firms. Similarly, the electronics firms did not have such a good knowledge of the requirements of commerce and industry for calculating machines as the business machine manufacturers. The result was natural enough—a pooling of resources. Firms tended to pair off like partners at a dance—National Cash Register with Elliott, British Tabulating with G.E.C., Powers-Samas with Ferranti, and so on—and arranged things so that the electronics firm tended to concentrate on the electronic circuitry and the business machine firm on the mechanical input and output equipment and general marketing of installations.*

This culminated in 1958 in the first electronic computer exhibition to be held in Britain—actually the first of its kind in Europe. Significantly it was organized on a joint basis by the two official trade associations representing the firms concerned, the Electronic Engineering Association and the Office Appliance and Business Equipment Trades Association. In fact this combination might be said to represent the electronic computer industry in Britain today.

---

* **British Tabulating and Powers-Samas have now combined to form International Computers and Tabulators.**

Thus, after many years of separate development, the two main types of computing machine—commercial and academic—are coming together again on the common ground of electronic techniques. Undoubtedly this is leading to a most fruitful exchange of ideas between the technicians of both branches. Of course, different kinds of computers and calculators will continue to be made in different ways for different purposes, but there will always be this common basis of electronic techniques, and already the title of " computer engineer " is being applied to people working on them. As the technique of electronics itself develops, so will the changes be felt throughout the whole field of computing. At the moment, for example, there is a trend towards the replacement of thermionic valves by transistors in electronic apparatus, and this is having a considerable effect in the computing field.

One could speculate a great deal on the possible future of the digital computer, seeing it grow more and more versatile and able to cope with subtle and complex problems of the kind that can be solved only by the human brain at present. Already computers have shown themselves capable of playing games, translating languages, recognizing patterns and even " learning " from experience. Whether they can be said to " think " is another matter, but there is no doubt about the fact that experimental machines with computer-like characteristics have been built to produce types of behaviour that, in animals or human beings, would definitely be described as " intelligent ". However, such matters are best left for discussion in a later chapter, after some of the more immediate aspects of computers have been considered.

2

GENERAL PRINCIPLES OF COMPUTING

IN the first chapter it was mentioned that all computers can be divided into two distinct kinds, ***analogue*** and ***digital.*** Analogue computers represent numbers or quantities by something else—length along a scale, angle of rotation, voltage, magnetic flux, etc. Probably the best-known example is the slide-rule. Distance along the rule is made to represent the logarithm of a number, and since adding the logs of two numbers gives the log of the product, multiplication can be done simply by adding together lengths of scale. Division is done by subtracting lengths; taking square roots by halving them, and so on. If the scales of a slide-rule were uniformly divided instead of logarithmic, it could be used for simple addition and subtraction, but that would hardly be worth while except if it were used as part of a more complex machine.

The predictor is a fairly complicated analogue computer, designed for the particular job of rapidly working out gunnery settings. A mechanical analogue computer intended for educational purposes is shown on p. 29. Here the numbers or quantities are represented by linear displacements of the moving parts.

The best-known digital computer (apart from one's fingers) is the abacus—that frame with beads strung on wires, supplied to infants with the hopeful intention of teaching them arithmetic under the guise of entertainment. That it is capable of more than this was shown in a contest between a Japanese virtuoso of the instrument and an American exponent of a commercial desk calculator—in which the Japanese won every time.

The essential difference between the two classes is that digits or numbers are separate isolated things—the mathematical term is *discrete*—whereas the analogues are continuous. For example, one-third cannot be expressed by any finite number (being 0·333333 . . . for ever), but one-third of a distance is possible. In practice, however, this does not mean that the analogue principle leads to greater accuracy; quite the contrary, for a scale of manageable length has to be very carefully made if it is to be read reliably to even three significant figures, whereas digital systems can be made to handle figures with any desired number of " places " simply by duplicating components, like valves or relays, which in themselves call for no unusual precision of workmanship. The automatic telephone is a digital system, and the seven twists given (in the London area) to its mass-produced dial, generating seven groups of up to ten pulses, could be used to select any one of ten million different numbers. An analogue scale capable of being read unmistakably to one in ten million would be rather a formidable proposition for the instrument maker!

The limit of precision for reasonably economical manufacture is in the region of one in a thousand, and if one's needs come within that (as most everyday engineering calculations do) an analogue device is generally the cheaper way of meeting them. A digital computer capable of doing the work of a slide-rule would be much more expensive. But the cost of obtaining greater precision by finer workmanship goes up very steeply and soon soars beyond all possibility, whereas the curve for digital systems rises slowly, so in practice they are always adopted for really precise work.

Another way in which computers can be classified is according to the types of work they take off one's hands and brain. To follow this you will have to give a little thought to what you do when you perform a calculation without external aids; that is to say mentally, because pencil and paper are very definitely external aids. Suppose you have to work out how much a dozen articles at 19s. each are going to cost you.

There are various ways of doing it, of course. The most obvious is to multiply 12 by 19 and if necessary divide by 20 to bring to pounds; but most of us never learnt the nineteen-times table, and find it rather a strain to do a long multiplication in our heads—the reason being that we may forget one partial result before the other is ready to add to it. But noting that 19 is one less than 20 we may multiply 12 by 20 and then subtract 12. We do the multiplication in two stages; first by 2 and then by 10. The first stage comes within the scope of the multiplication tables held in our memories, so after planning the whole sequence as above we draw on memory for the fact that $2 \times 12 = 24$. The advantage of the selected procedure is now apparent; there is no need to leave " 24 " somewhere in our minds while we get on with another operation. Although 24×10 is beyond the twelve-times table, we draw on another file in our memory for the fact that multiplying by 10 in the decimal system of numbers merely requires a movement of one " place " or digit to the left, a " 0 " being added on the right to show this. Subtracting 12 from 240 is perhaps the most difficult task, and it may be interesting to consider exactly how we do perform this part of the job. Do we first take away 10, reducing 240 to 230, and then another 10, reducing it to 220 and (making use of the memorized fact that $10 - 2 = 8$) add to the units column to give the answer 228s.?

Of course, if we had wanted the answer in pounds and shillings we could have eased the work by bearing this in mind from the very beginning, converting 12×20 shillings direct into £12 and then converting £1 of this into shillings so as to be able to deduct 12s.

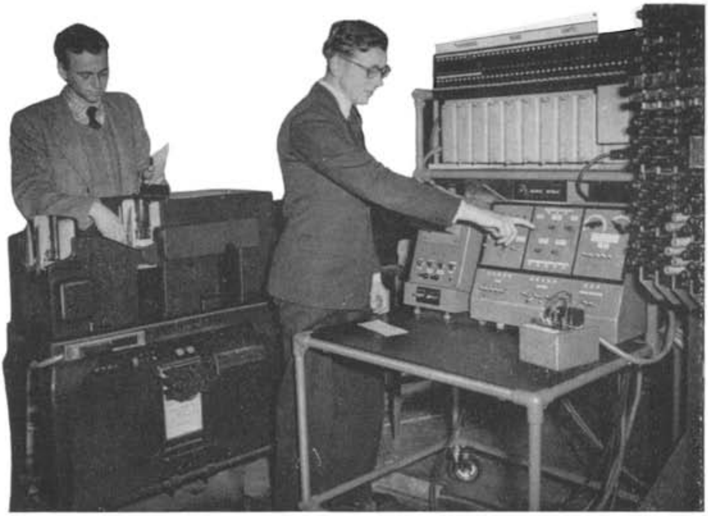
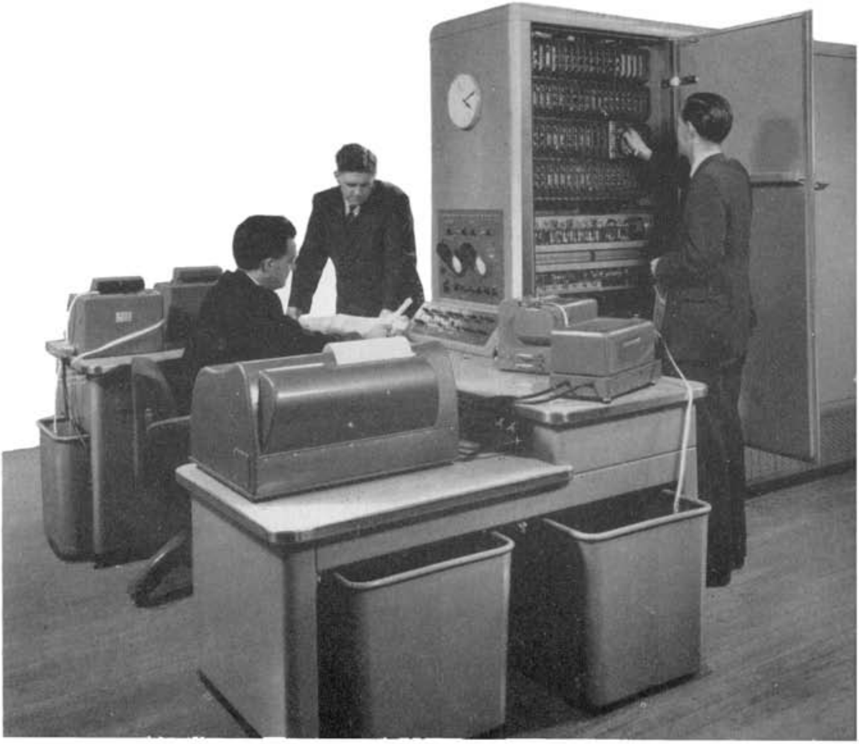


Plate 1 (a). Input and control equipment of an early British digital computer—the pilot model of the ACE at the National Physical Laboratory

Plate 1 (b). A modern digital computer built by Ferranti on the packaged-circuit principle, using standardized plug-in units



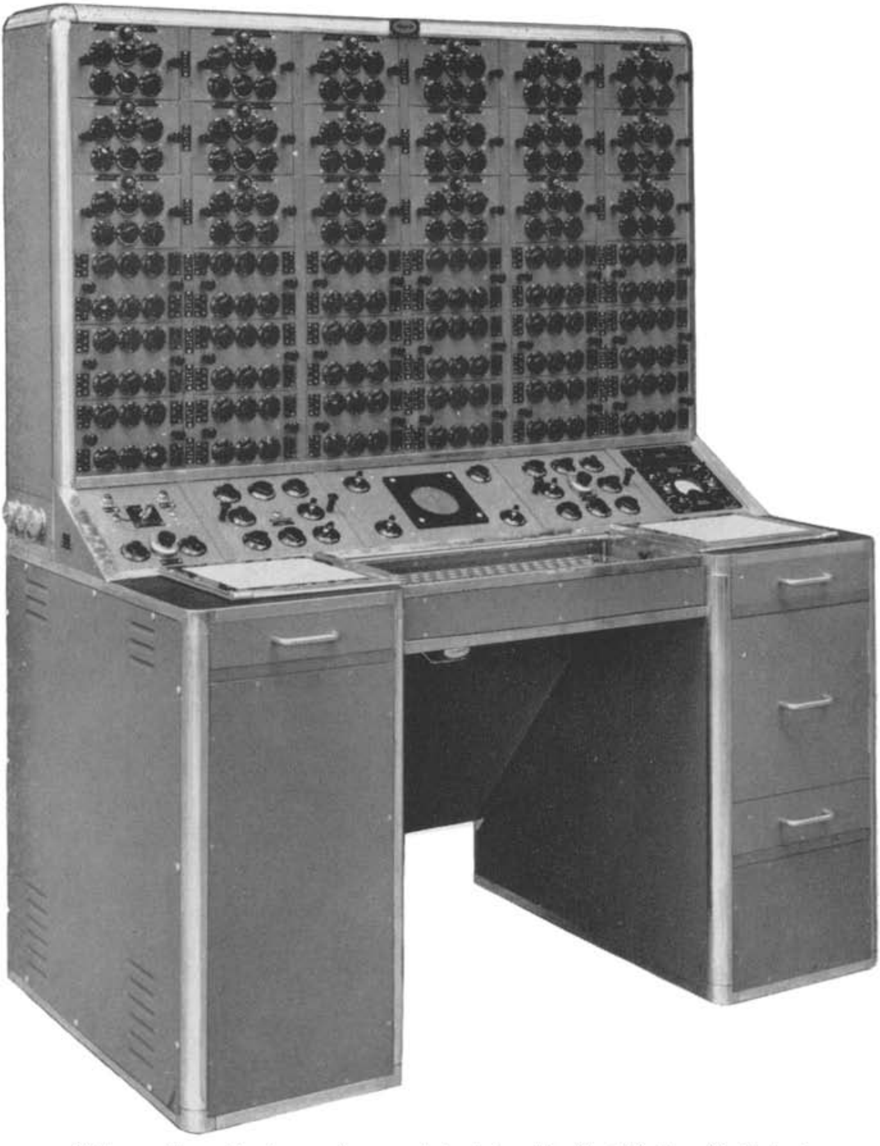


Plate 2. General-purpose analogue computer designed by Short Brothers & Harland. The computing elements are constructed as small detachable units

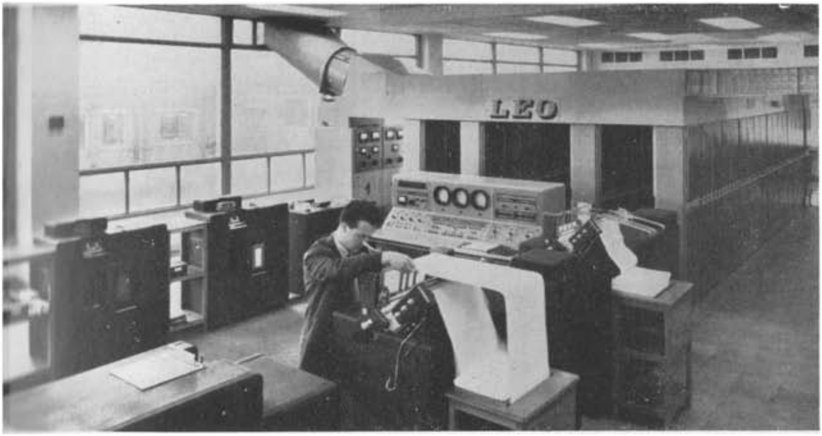


Plate 3 (a). Large digital computing installation by Leo Computers

Plate 3 (b). The ACE digital computer at the National Physical Laboratory

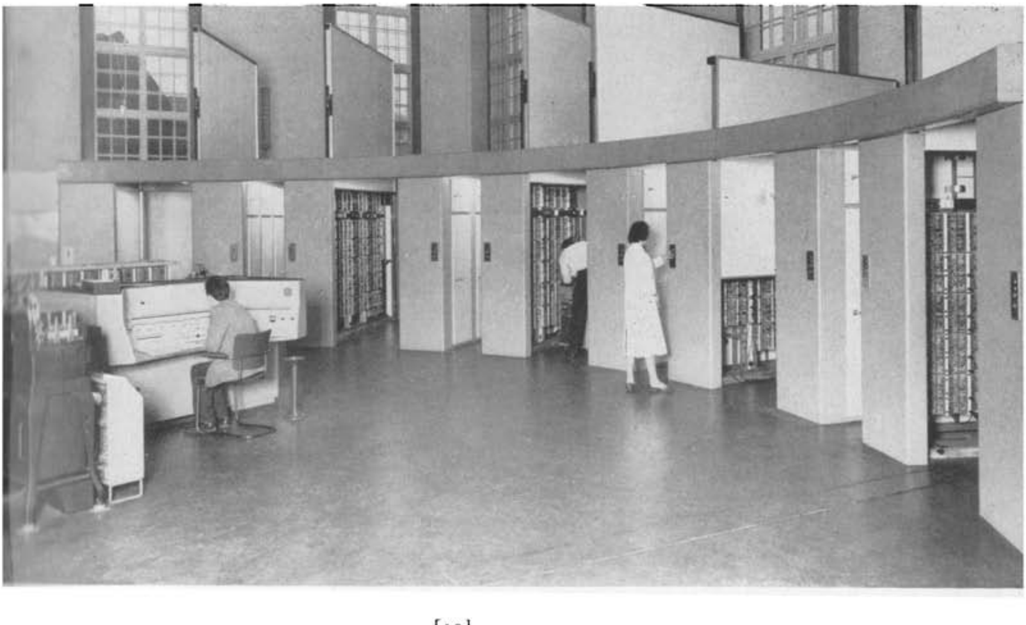
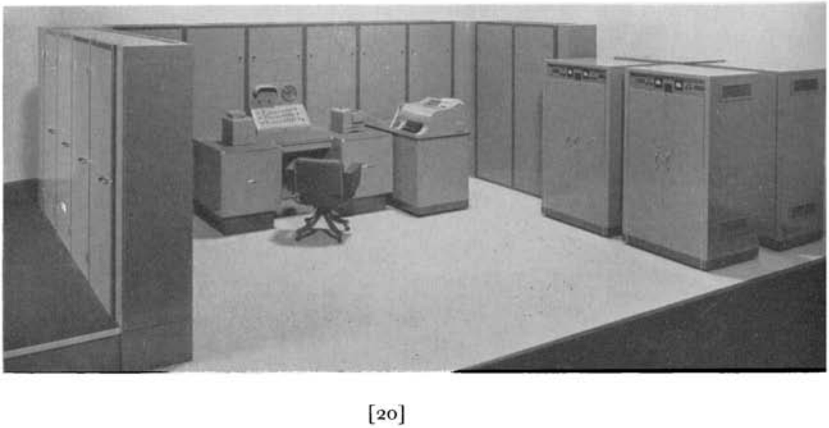


Plate 4 (a). A factory production line for medium-size digital computers

Plate 4 (b). The Mercury high-speed digital computer designed by Ferranti for scientific work



It has been necessary to go into this simple piece of mental arithmetic at such length because it illustrates the three main departments of any computing system. There are one or (usually) more arithmetic operations, such as addition, multiplication, etc. Then there is storage or " memory " of two kinds—permanent stores for such things as multiplication tables, and temporary stores or parking places for partial results. Then there is the planning and carrying out of the whole computation in sequence.

It is as a temporary store that a piece of paper is such a valuable aid. In long multiplication and division the results of the separate single-figure operations can be recorded on it until we are ready to add them together. Even in simple addition it is difficult to carry in one's head the figures that one is not actually dealing with at the moment.

Paper and pencil leave the brain to supply the permanent stores of information, but a slide-rule takes over most of that job too. It also reduces the number of separate operations by dealing with two- or three-figure numbers at one time. So it needs less temporary storage, and most of what is required is provided by its cursor. It is only when multiplications and divisions are interspersed with additions that there is need to record partial results on paper. If a large number of similar calculations has to be done on a slide-rule it is worth while spending some time and thought on sequence-planning, to accomplish each with the fewest movements of slides and cursor. For instance, if a formula such as $\frac{p \times q \times r \times s}{x \times y \times z}$ has to be worked out for numerous values of p, q, r, etc., it would be foolish to do all the multiplications first and then all the divisions. It is best to divide p by x and multiply by q in one movement, and so on.

The corresponding digital aid is the desk calculator. There are many varieties, but however they may be arranged mechanically the basic element is a counter. A motor car contains both analogue and digital instruments in one assembly—the speedometer uses pointer angle as the analogue of speed, and the mileage indicator, which is a counter, reckons distance in terms of figures. The usual decimal counter has for each digit a wheel bearing the ten figures 0 to 9 on its circumference. The wheel is arranged to be moved round one-tenth of a revolution for each unit of number. To count beyond 9 it is necessary to have a second wheel, for the tens digit, which the first wheel pushes forward one step whenever it turns from 9 to 0. Similarly for hundreds, thousands, and so on *ad lib*.

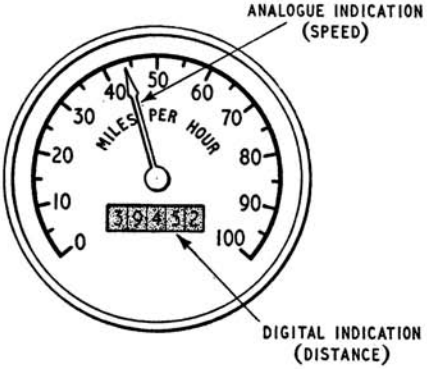


Fig. 2.1. This familiar instrument in a motor car indicates the speed by an analogue method and the distance by a digital method

Although so simple, this device manages to do the " carrying " operation automatically at the same time as the addition. For instance, if the units wheel standing at 8 is pushed forward five steps it not only registers the correct units digit resulting from 8 + 5 but it also carries 1 to the tens position. To do the same thing with electrical components, however, is not as easy as it may seem. The carrying may have to be done as a separate operation.

Neither the slide-rule nor the desk calculator is automatic. The operator not only has to plan the whole sequence of operations; he has to carry them out. Modern digital computers, however, are automatic, or sequence controlled, so that the operator has only to do the initial planning or " programming ", and the machine does the rest. As mentioned in the first chapter, they are also capable of varying the sequence in the light of the partial results obtained, and in this lies an important advantage. To take a simple example, the machine could be made to repeat a sequence of operations until the answer became negative, or equal to a stated number, or a maximum or minimum, and then start a different series of operations.

So the modern automatic computer does everything except provide the problem and the programme. It is hardly fair to call it a " brain ", however, because most of the real brain-work goes into drawing up the orders so that the machine can carry them out. What the machine does is to take over the laborious and tedious routine operations, and (in the electronic types) at thousands of times the possible human speed and with far less likelihood of errors. Even a mathematician needs several weeks to learn how to instruct one of the big electronic machines. The problem is that the instructions have to be made absolutely complete and unambiguous in every detail, and yet expressed in terms the machine can " understand ".

Even the instructions for multiplying one number by another are quite considerable. But a great deal of effort can be saved by building frequently used sequences (called sub-routines) into the machine. This is rather like what is done when a complicated plan (such as a military offensive) has to be put into effect at a moment's notice—the order can be reduced to " Operate Plan B " or a simple code word or number, so long as the details have been cut and dried in advance. A code number fed in brings into operation a previously prepared and stored record of the sequence to be followed by the machine.

Now to the actual electrical and electronic techniques employed in computers. First, the analogue kind. One way of representing the numbers in the computation is by voltages. It is easy to add or subtract them by rotation of a potentiometer, forwards or backwards. Sines and cosines are obtainable by a suitably shaped potentiometer, or (with a.c.) by the angular position of a coil in an alternating field. The output of a potentiometer is proportional to its setting multiplied by the voltage applied to the input, so if both are under control it is possible to multiply. Another method of multiplication is to represent the variables by the currents in two coils; the force between them is proportional to the product.

Such devices are mechanical or electro-mechanical in operation, whereas in this book, of course, we are mainly concerned with electronic methods of computation. Most electronic analogue computers work on the principle of representing numerical values by voltages and operating on these voltages with devices incorporating valves and cathode-ray tubes. Adding, subtracting, multiplying, sign-reversing and integrating are some of the operations which are commonly performed.

There would be little point in using an analogue computer for calculations which required only a single, fixed solution from these operations. It would be simpler to use just pencil and paper. When, however, it is necessary to do calculations on numerical values which are varying continuously with time, the machine becomes particularly valuable as a labour-saving device. Because the analogue computer can produce continuously varying solutions just as easily as single, fixed solutions (in contrast to pencil-and-paper methods) it is used principally for solving dynamic problems in science and engineering. For this reason the numerical value or values fed in, and the solution obtained at the output, usually take the form of voltages varying with time. The solution, therefore, is not just a steady voltage representing some number, as in the simplest kind of analogue computer, but an electrical waveform (i.e., a graph of voltage varying with time) and is usually presented on a cathode-ray tube or on the chart of a pen recorder.

In the electronic analogue computer the process of integration is performed by a circuit which utilizes the build-up of charge in a capacitor. This process is, in fact, used very extensively in an important class of analogue computers known as " differential analysers " (mentioned in Chapter 1). These machines are used for solving differential equations, although, as somebody has pointed out, the way they actually work would be more accurately described by " integrating synthesisers ". By the nature of their apparatus they can only do the integrating with respect to time, but in practice this is not a serious limitation because most of the calculations on dynamic systems in fact require this type of operation.

It is important to realize, however, that the differential analyser, or any other type of analogue computer, does not produce general solutions to equations to which numerical values can be given afterwards. By the nature of the analogue principle it can operate only on actual numerical values which can be represented by voltages, and so will only give solutions to particular problems. The digital machine is similarly restricted. It is only an arithmetic device, and if one wishes to use it for solving mathematical problems one has to employ various tricks for putting the mathematical equations into numerical form.

In digital computers, as already explained, numbers are represented in terms of their digits by separate isolated things like teeth on a wheel or electrical pulses. But this is not always done on the basis of our familiar decimal system. We get so completely used to our system at such an early age that we tend to regard counting in tens almost as a law of nature, and are quite horrified when someone proposes that it would be better to count in twelves and make the sign " 100 " mean the number we now call 144. Yet there is no reason more binding than widespread custom for making ten the number at which to change from single to double figures, unless it be that most people's fingers and thumbs add up to ten. As the pioneer who invented the dozen probably realized, twelve is a better number than ten on which to base a system, because it is divisible by two, three, four and six, instead of only two and five.

But that is by the way. The point is that, although our familiar decimal (or scale-of-ten) system is used so universally that it has never occurred to most people that there could be any other, we are free to base a system on any number we like. If we adopted the duodecimal or scale-of-twelve system it would be necessary to have two new single figures to stand for ten and eleven, so that the first double figure (10) could mean twelve. On the other hand, a scale of fewer than ten puts some of our present figures out of a job. For the number of the scale is really the number of different values or levels that can be written in single figures. In the scale of ten there are ten: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. The smallest scale that can be used is two, and it requires only two figures, 0 and 1. This is, in fact, the scale that is used in most digital computers today. Why is this so?

Imagine you are starting to invent an electronic calculating machine. You are using valves so as to get quicker action than is possible with mechanical contrivances. For convenience in reading the answers you connect a milliammeter scaled in whole numbers from 0 to 9 in the anode circuit of the valve dealing with the " units ", and similarly for tens, hundreds, etc. The valve is initially biased beyond cut-off, so that the milliammeter reads 0. By means of cunningly devised circuitry you arrange that when the pulses or other signals representing units arrive each one steps up the anode current by 1 mA, and directly 9 mA is exceeded the " units " valve flops back to the start and at the same time sends a signal to the " tens " valve to make it read 1. This is what mileage indicators and other counters do mechanically.

Before long you would probably decide that it might be a waste of time going any further in working out the details of such a system, because it would need a good deal of setting up to make all the milliammeters read the whole numbers correctly all over their scales, and it would also be necessary to have all their supply voltages very effectively stabilized, otherwise the readings would be upset by any mains fluctuations. And even then one could never be sure that the slope of at least one of the valves had not drifted half a milliamp or more—enough to raise a doubt as to which number was meant.

For the really large-scale calculations that justify the expense of electronic computing machines it would be out of the question to allow any risk of slipping a whole number here and there, perhaps in the billions column. A valve is all very well, but one is unwise to rely on it to click with mathematical precision and mechanical rigidity into ten different levels on its characteristic curve. The fact that it actually is curved and not straight is another difficulty. And it is taking a risk to expect the triggering action to distinguish infallibly between 9 mA and 10 mA.

It is not, however, asking too much of a valve to expect it to distinguish between " off " and " on ". The bias for " no current " can be made as negative as is necessary to ensure such a condition no matter how much the valve characteristics or the voltages are off-centre; and so long as the current is large enough in the other condition to be recognizable as a current, it does not matter (within reason) how much more than the minimum it is. So non-linearity is of no account whatever, nor are drifting voltages or valve characteristics, so long as the valves can still pass any current at all. On this basis the whole thing is as reliable as it possibly can be.

And that is where the scale of two, or binary scale, comes in. Since it has only two varieties of figure, 0 and 1, its whole scale is fully represented by the two conditions " off " and " on ". And yet the system can count up to any number, the only inconvenience (apart from its unfamiliarity) being the larger number of figures or digits, and hence valves.

Let us see how it works out. *Nought* is, of course, denoted by 0 everywhere. And *one* is denoted by 1. There is no such thing as " 2 ", so adding another *one* means carrying to the next—the " twos "—column, where a 1 appears, and the 1 in the units column is cancelled. This procedure is exactly what we do in the decimal system when adding *one* to *nine*.

Adding another *one* brings 1 back into the units column, so we have " 11 ". (In the decimal system that symbol denotes one *ten* plus one *one* = eleven; in the binary system it denotes one *two* plus one *one*.) Adding the fourth *one* clears the units column again, as the *two* there is carried to the twos column; and as that makes two twos it also is cleared and carried to the next—the " fours " column. Result: 100 (to be read as " one–0–0 "). And so on, as can be seen from the table on page 31.

By now it will probably be realized that the next column after the " fours " shows the absence or presence of an eight in the total; the next, a sixteen, and so on. For example: 1001101 in the binary notation would mean, working from right to left $(1 \times 2^0) + (0 \times 2^1) + (1 \times 2^2) + (1 \times 2^3) + (0 \times 2^4) + (0 \times 2^5) + (1 \times 2^6)$. This comes to $(1 \times 1) + (0 \times 2) + (1 \times 4) + (1 \times 8) + (0 \times 16) + (0 \times 32) + (1 \times 64) = 1 + 4 + 8 + 64 = 77$. (Note that any number to the power 0 is 1.)

Thus, whereas decimal numbers are formed on the base ten and the decimal places (working from right to left) have values 10^0, 10^1, 10^2, 10^3, 10^4, etc., the binary code is formed on the base 2 and the corresponding positions (again working from right to left) have values 2^0, 2^1, 2^2, 2^3, 2^4, etc. Some examples of decimal numbers and their equivalents in binary notation are given in the table shown on page 31.

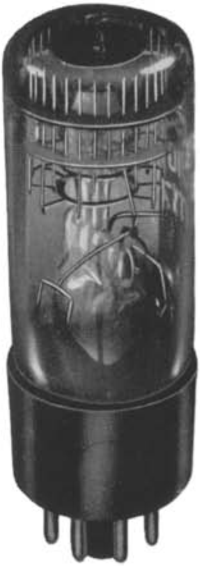


Plate 5 (a)—left. One form of decimal counting tube (Dekatron) in which a glow discharge is shifted step by step round a circle of wire electrodes

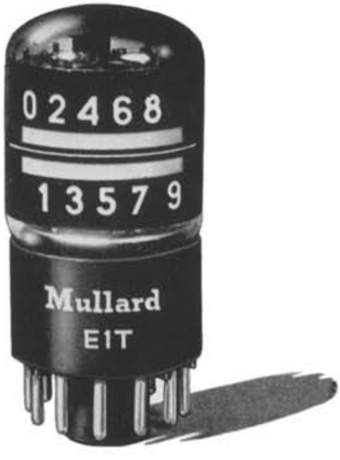


Plate 5 (b)—right. Another decimal counting tube, operating by deflection of an electron beam into different angular positions

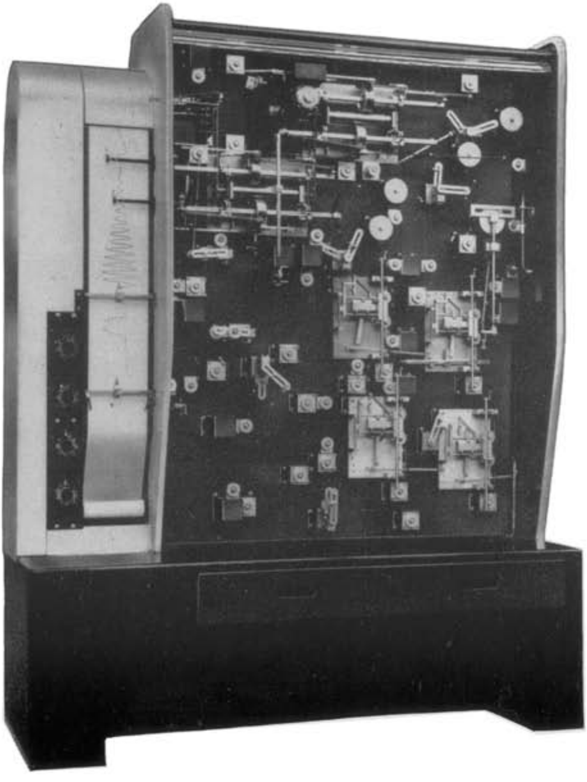


Plate 5 (c). Example of a mechanical analogue computer built by Air Trainers for instructional purposes

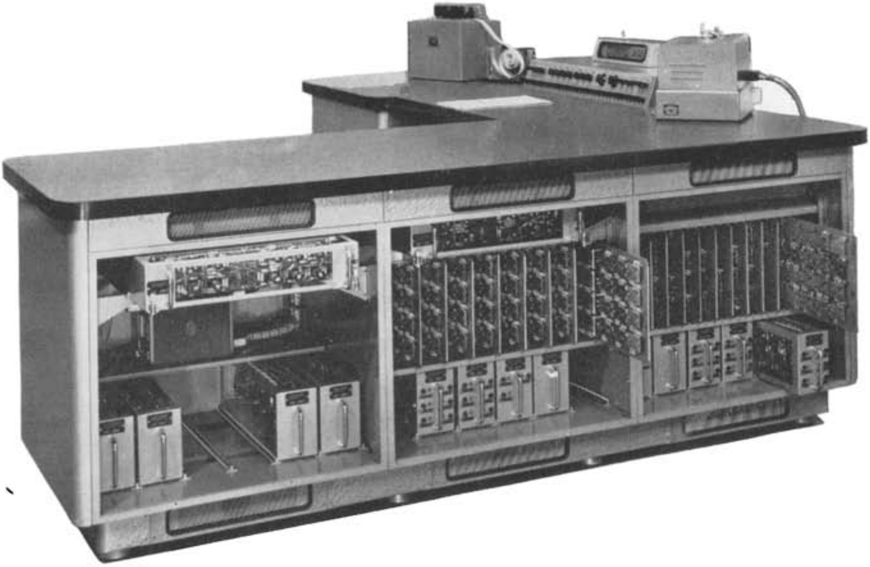
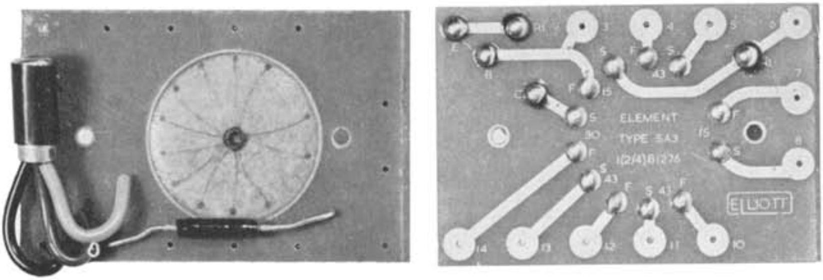


Plate 6 (a). The Elliott 802 transistor/magnetic-core digital computer intended for process control applications

Plate 6 (b) and (c). Back and front views of a transistor/magnetic-core computing element used in the Elliott 802 machine above



Decimal number	*Binary number*	*Decimal number*	*Binary number*
0	0	11	1011
1	1	12	1100
2	10	13	1101
3	11	14	1110
4	100	15	1111
5	101	16	10000
6	110	17	10001
7	111	18	10010
8	1000	19	10011
9	1001	20	10100
10	1010	and so on	

Incidentally, decimal numbers can be converted into binary form by successively dividing by 2 and noting the remainders and the final quotient, all of which must be either 0 or 1. For example, to convert 37 the procedure would be: $37 \div 2 = 18$ (remainder 1); $18 \div 2 = 9$ (remainder 0); $9 \div 2 = 4$ (remainder 1); $4 \div 2 = 2$ (remainder 0); $2 \div 2 = 1$ (remainder 0). Working from the last quotient, 1, forwards, this gives 100101.

Though it would be very muddling to introduce the duodecimal or binary or any other non-decimal scale of counting into everyday life, we have seen that the binary system has distinct advantages for electronic computing. Reducing the number of figures, or distinguishable states, from ten to two is such an enormous simplification that it is well worth having even though it means about three times as many digits, and also "translating" from and into decimal notation. For example, if numbers up to 9,999,999 are to be handled, it is easier to do it with twenty-four 2-way (on-off) switches than with seven 10-way switches.

The binary system of arithmetic, then, is used in most digital computers as a matter of practical convenience—because the electronic valve gives its most reliable operation as a two-state device. At the same time, however, there is a specialized class of electron tubes used widely for electronic counting which is particularly suitable for decimal operation because the valves have the equivalent of ten stable states. These ten states are actually the angular positions which can be taken up by an electronic conducting path in a system of electrodes. One example is the Dekatron tube (p. 29, Plate 5[a]), in which a glow discharge can be moved round a circle of ten cathodes (grouped about a common anode) by applying pulse signals to a system of control electrodes. In another type of tube an electron beam is deflected to any one of ten positions, again by voltage control signals (p. 29, Plate 5[b]).

Tubes of this kind are already being used to some extent in digital computers but it is not likely that they will cause an appreciable swing-over from binary to decimal arithmetic. At the moment the computers working on scale-of-two principles are by far the largest class, and for this reason the later chapters on digital computer circuitry are devoted entirely to binary techniques.

It is worth mentioning, however, that quite a number of small digital computers use a form of decimal arithmetic in which the individual digits of the decimal numbers are represented in binary notation. This is called binary-coded decimal. Thus the decimal number 57 would not be represented in the pure binary form of 111001 but as 101, 111. This system preserves the advantages of working with two-state electronic devices and at the same time keeps the numbers in a more convenient form. It also avoids the complicated procedures necessary for converting into and out of pure binary notation. Unfortunately, however, the system requires a good deal more electronic equipment than is necessary for pure binary arithmetic.

Of course it is possible to devise other binary coding systems by which numbers (or even letters) can be represented by combinations of the two digits. Readers may, in fact, have seen references to such things as the 2421 code or the excess-three code in connection with computers or other digital systems. These, however, tend to be essentially *coding* systems, in the full sense of the word, rather than direct systems of numbering like pure binary notation. The individual digits and the patterns in which they are grouped are arbitrarily chosen to have special significances for particular types of application, whereas the digits of the pure binary system have straightforward numerical values. For the sake of simplicity this book will deal only with the commonly used pure binary and binary-coded decimal systems.

3

ANALOGUE COMPUTING CIRCUITS—I

A GOOD deal of analogue computing work has been done in the past by means of electro-mechanical instruments. Nowadays there is a very pronounced trend towards purely electronic techniques, which have the great advantage of requiring no moving parts. The modern electronic analogue computer does not therefore have to be a precision engineering job, and can be manufactured at reasonable cost from standard mass-produced electronic components with almost the same facility as a radio or television set. It is more compact and convenient to handle than the electro-mechanical computer, and it has a greater flexibility in the time scale. Results can be produced fast enough to be displayed on a cathode-ray tube or slow enough to be plotted by hand on paper.

This does not mean, however, that the electro-mechanical computer is completely outmoded. It still has advantages in certain applications; for example, in control systems where a computing function has to be " built in " and integrated with mechanical equipment. Here the introduction of electronic circuits might well prove a needless complication.

Moreover, it is not at all unusual to find pieces of electro-mechanical apparatus in the modern electronic analogue computers. It is purely a matter of convenience and expediency: sometimes the electro-mechanical method is better and sometimes the electronic method.

As already mentioned in Chapter 2, electronic analogue computers use voltages to represent numbers and values—for example, 50 V might represent a mechanical pressure of 50 lb—and these voltages operate in circuits containing principally resistors, capacitors and thermionic valves or transistors. To take a simple example, if a voltage of 10 V is applied to the element of a linear potentiometer, as in Fig. 3.1 (a), and the wiper is positioned half-way along the element, then the voltage obtained on the wiper is 5 V. In other words, the original voltage is multiplied by 0·5—and, of course, it is obvious that the original voltage can be multiplied by any fraction from 0 to 1, depending on the position of the wiper. If it were necessary to multiply by a number greater than 1, one solution might be to use an amplifier with a known degree of amplification or "gain". Thus if an amplifier with a gain of 10 : 1 were connected to the potentiometer as shown at Fig. 3.1 (b), the final result would be to multiply the original voltage by 5, giving an output of 50 V. (In practice amplifiers are hardly ever used in this way because of the difficulty of keeping the gain at a prescribed figure with sufficient accuracy.)

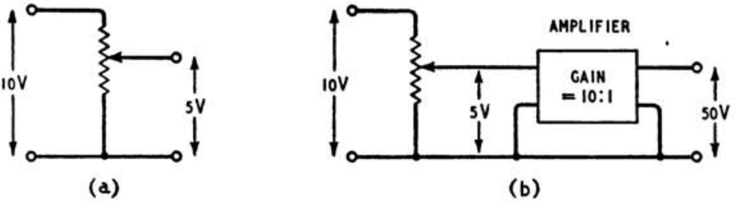


Fig. 3.1. A potentiometer used for multiplication, here × 0·5 (a); by using an amplifier after the potentiometer it is possible to multiply by numbers greater than 1 (b)

For adding numbers together, the most straightforward analogue method is to connect the voltage sources representing them in series. This, however, is not always very

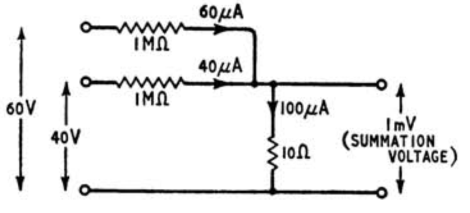


Fig. 3.2. Principle of adding numbers by summation of currents in a small output resistor

easy to do in practice because it means that all but one of the sources have to be " floating " or isolated from earth. It is more convenient to work on the principle of summing the currents produced by the voltages. For example, in Fig. 3.2 the currents flowing through the two 1-MΩ input resistors are proportional to the voltages applied. These two currents both flow through the same 10-Ω output resistor and so produce a voltage across it proportional to their sum. If the 60-V input were reduced to, say, 30 V, the total current through the 10-Ω resistor would be 70 μA and the output 0·7 mV.

It will be seen, then, that the output voltage produced is always equal to a certain fraction (1/100,000) of the sum of the input voltages—and the same would apply if other input voltages were connected through corresponding 1-MΩ resistors. Subtraction can be achieved by adding a negative voltage, while it is possible to multiply any of the input voltages by a constant factor by altering the value of the corresponding resistor. For example, if the 1-MΩ resistor in the 60-V input were halved the current in that branch would be doubled to 120 μA and the output voltage would become 1·6 mV—corresponding to the addition of 120 V to 40 V instead of 60 V to 40 V.

Actually the summation by this method is only accurate if the output resistor is very small compared with the input resistors (as it is in this particular example) so that its value cannot have a substantial influence on the currents flowing in the individual input branches. This means that the output voltage is really too low for convenience in practical circuits. The possibility of using an amplifier comes to mind for multiplying the output voltage the required number of times ($10^5 : 1$ in Fig. 3.2) but again there is the problem of stabilizing the gain to within sufficiently accurate limits.

One method of stabilizing the gain of an amplifier is the well-known technique of applying negative feedback. A certain proportion of the amplifier output is fed back to the input circuit in such a way that it tends to oppose any fluctuations in amplifier gain. If such a fluctuation causes, say, an increase in the output voltage, this increase, when fed back to the input, tends to oppose the effect of the input voltage and so produces a decrease in amplifier output. The result of this compensatory action is, of course, that the output voltage is maintained constant for a constant input voltage.

There are various ways of applying negative feedback to amplifiers, but in analogue computing it is done as shown in Fig. 3.3. A high-value resistance, R, is connected from the output of the amplifier to its input, and the *negative* nature of the feedback is achieved by virtue of the fact that the amplifier has an odd number of valve stages (say, three) so that the output voltage has the opposite

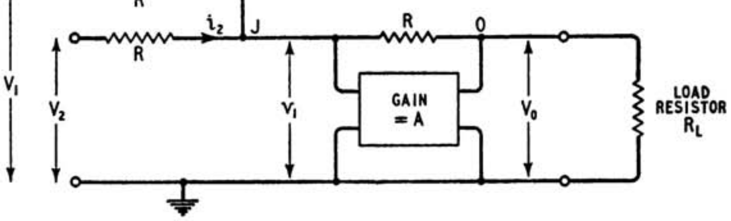


Fig. 3.3. Use of amplifier with negative feedback resistor for summation of voltages

polarity to that of the input voltage. The amplifier has to be of rather special design, too, in that it must be capable of amplifying steady voltages or voltages with slow fluctuations—as distinct from the normal sort of audio amplifier which only deals with fluctuating voltages of frequencies above 20 c/s or so. It is therefore usually described as a " direct current amplifier " or, simply, d.c. amplifier. The actual gain is usually very large—perhaps 50,000 : 1 or even higher. (More will be said about these d.c. amplifiers in the next chapter.)

It will be seen that the amplifier and feedback resistor are connected to an arrangement for adding similar to that in Fig. 3.2, except that there is no low-value resistor like the 10-Ω one for current summation. The feedback resistor, moreover, has the same value as the two input resistors. But in this arrangement the feedback resistor does not merely act as a means of stabilizing the amplifier gain. It plays a much more fundamental role in the whole process of adding.

Supposing that positive input voltages of, say, 60 V and 40 V are applied to the input terminals (as in Fig. 3.2), in the normal course of events the junction point J would go positive in potential. If the amplifier had no feedback its output would be an extremely high negative voltage (bearing in mind its polarity reversal and high gain) and this voltage would be considerably greater than the − 100 V that one would expect across R_L from the summation process. What the feedback resistor does, in fact, is to lower the positive potential of the junction point J by virtue of the fact that its right-hand terminal, O, is connected to a source of negative voltage—provided by the amplifier output.

One can see that if the input resistors are connected to positive potentials and the right-hand end of the feedback resistor to a negative potential, then one has a form of potential divider in which the point J must be at an intermediate potential. In fact, it works out that J is extremely near to earth potential, being only slightly above it by the small voltage v_1 (positive in this case) which forms the input of the amplifier. And when v_1 (perhaps a few millivolts) is multiplied by the gain of the amplifier with feedback, the output comes to a voltage which is the sum of the input voltages, only with negative polarity (in this case $-$ 100 V). Why it should necessarily work out like this will be shown later.

If the point J is maintained virtually at earth potential by the action of the amplifier and feedback resistor, then the situation of the two input resistors becomes closely comparable with the equivalent two resistors in Fig. 3.2. In other words, the low voltage between J and earth is tantamount to having a low resistance connected across these points like the 10 Ω in Fig. 3.2. It was clear from Fig. 3.2 that the lower the value of the common resistor—and hence the voltage developed across it—then the more accurately were the currents in the 1-MΩ resistors proportional to the voltages applied to them. A similar situation prevails in Fig. 3.3. Only if the point J is virtually at earth potential do the currents in the input resistors, i_1 and i_2, correspond to the applied voltages V_1 and V_2. But in this case, of course, the total current $i_1 + i_2$ does not flow through an actual small shunt resistor but through the high-value feedback resistor R.

Supposing then that $R = 1$ MΩ, V_1 and V_2 are 60 V and 40 V, and i_1 and i_2 are 60 μA and 40 μA as in Fig. 3.2, it can be seen that the total current of 100 μA flowing through the feedback resistor will produce a voltage of $-$ 100 V at the output (remembering that point J is virtually at earth potential). Thus the output voltage is the sum of the two input voltages, only with reverse polarity. If one of the input voltages were increased, the point J would tend to go more positive. This would produce a greater negative output voltage from the amplifier at O and so bring the point J back towards earth potential (by the potential divider action). The system would settle down with J at almost earth potential and with the increased negative voltage at O corresponding to the additional voltage applied at the input. (Obviously the potential at J cannot be exactly at earth potential, otherwise v_1 would be zero and the amplifier would give no output.) The amplifier and feedback resistor, in fact, together form a self-adjusting system which is dependent on the amplifier having an extremely high gain.

It is quite simple to show that $V_0 = V_1 + V_2$ when the amplifier gain is very high by a straightforward algebraic proof. To begin with, one can see that

$$i_1 = \frac{V_1 - v_1}{R} \quad \text{and} \quad i_2 = \frac{V_2 - v_1}{R}$$

and therefore

$$i_1 + i_2 = \frac{V_1 + V_2 - 2v_1}{R}$$

At the same time one can see that the voltage developed across the feedback resistor, $V_0 - v_1$, is $(i_1 + i_2)R$, from which it is clear that $i_1 + i_2 = (V_0 - v_1)/R$. Now if we equate these two expressions for $i_1 + i_2$ we have:

$$\frac{V_0 - v_1}{R} = \frac{V_1 + V_2 - 2v_1}{R}$$

From this $V_0 = V_1 + V_2 - v_1$. Now v_1 is given by the output voltage V_0 divided by the amplifier gain A, so this expression becomes $V_0 = V_1 + V_2 - V_0/A$. But if the gain A is very large, the last term on the right-hand side becomes insignificant and we are left with

$$V_0 = V_1 + V_2$$

Of course it goes without saying that further input resistors can be connected to the point J, as indicated by the dotted lines in Fig. 3.3. and the same principles of addition by current summation will apply with three inputs or more. If only one input voltage is applied, say

V_1, then the circuit can be used for sign reversal, and $V_0 = -V_1$. Using such a sign reverser it is possible to do subtraction on the principle of adding negative quantities. Multiplication by constant factors can be achieved, as already mentioned, by altering the values of the input resistors. Apart from these arithmetical processes, there is one other useful operation which can be performed by a similar technique—integration with respect to time.

This process of integration with time is based on the building-up of a charge in a capacitor when a voltage is applied to it through a resistor. Fig. 3.4 (a) shows the basic circuit. So long as the input voltage V_1 is present the capacitor goes on charging and the voltage across it, V_0, continues to rise in proportion to the charge until it reaches the applied voltage. The rate of change of the charge with time is the current flowing into the capacitor. Thus in Fig. 3.4 the voltage V_0 is proportional to the time integral of the current sent through R by V_1.

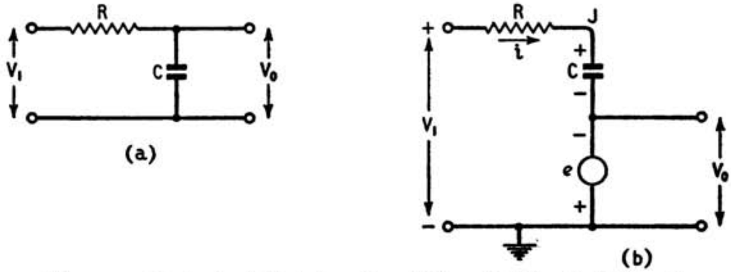


Fig. 3.4. Basic circuit for integration at (a); principle of offsetting the back voltage across C by a voltage generator (b)

This simple arrangement, however, does not provide what is necessary, for it integrates the input *current* to produce voltage, not the input voltage, and unfortunately the input current is not proportional to the input voltage. This is because the voltage which is building up across C progressively opposes the flow of input current through R and the current does not remain constant for a constant value of V_1. The result is that V_0 rises exponentially instead of linearly as it should if it were properly integrating the steady V_1. What is wanted is a system which will allow the capacitor to charge at a current proportional to the applied voltage. When V_1 is constant the current flowing into C must be constant.

This is usually done in analogue computing by offsetting the effect of the back voltage developed across C. The general principle is shown in Fig. 3.4 (b). A voltage generator e is placed in series with the capacitor and is arranged to produce between its terminals a voltage of the same magnitude as that developed across C but of opposite polarity. As a result the voltage across C and e taken together is always zero. This means that the point J is at earth potential (it is commonly known as a " virtual earth ") and from the point of view of the current i flowing through R this is equivalent to J being actually connected to earth. In other words the current is $i = V_1/R$ and is therefore strictly proportional to the input. Since the capacitor voltage is proportional to the time integral of this current, it is therefore also proportional to the time integral of the input voltage. The generator voltage, which provides the output, has been arranged to be the same as the capacitor voltage, only reversed in polarity, so the final result is that

$$V_0 \propto -\int V_1 \, dt$$

i.e., the output voltage is proportional to the time integral of the input voltage, which, of course, is the desired result.

The usual method of achieving the generator e in practice is shown in Fig. 3.5. It will be noticed that the circuit is similar to that of Fig. 3.3 except that the original feedback resistor has now been replaced by the capacitor C. Comparing the arrangement with Fig. 3.4 (b), the output terminals of the amplifier provide the generator voltage and are connected in the same way between the capacitor and earth. As before, V_0 is taken from the generator (i.e., amplifier) output. Point J is not exactly at earth potential but at a very small voltage, v_1, of the order of millivolts. When this is amplified it provides the " generator " voltage in series with C, and since the amplifier has an odd number of stages the output is reversed in polarity with respect to the input and so given the same series-opposing arrangement as in Fig. 3.4 (b).

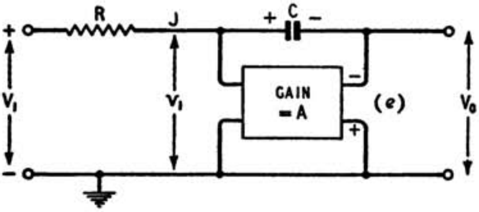


Fig. 3.5. Usual form of an integrating circuit, the amplifier being the means of achieving the voltage generator of Fig. 3.4 (b)

If a positive voltage is applied to the input (V_1) as shown, the tendency of point J to rise in potential causes the amplifier to drive the right-hand plate of C in a negative direction with respect to earth. This brings point J down again towards zero, and in fact the system automatically adjusts itself in this manner so that point J settles down to almost earth potential except for the small voltage v_1 necessary to provide an input for the amplifier.

The actual relationship of output to input must, of course, take into account the values of C and R, and a simple algebraic proof, similar to that applied to the adder, shows that

$$V_0 = -\frac{1}{CR}\int V_1\,dt$$

To take a simple example, if R were 1 MΩ and C were

1 μF and a steady voltage of + 1 V were applied to the input, the output voltage V_0 would increase linearly at a rate of − 1 V per second until the amplifier became overloaded.

Differentiation can also be achieved, working from the basic circuit in Fig. 3.6. This depends on the fact that the current passed by the capacitor is proportional to the rate of change of the applied voltage V_1, so that the voltage V_0 developed across R is also proportional to the rate of change of V_1. The principle, however, is not very often used in amplifier circuits of the Fig. 3.3 and 3.5 type. This is because the differentiator, in so far as it responds to rates of change, is very susceptible to any irregularities in the input voltage or spurious voltages produced by electrical interference. It is also prone to parasitic oscillation. For this reason, in this type of analogue computing technique differential equations are usually arranged so that they can be solved by integrating circuits.

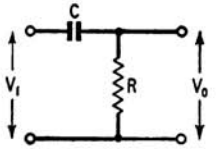


Fig. 3.6. Basic circuit for differentiation, but not used very much for analogue computing

4

ANALOGUE COMPUTING CIRCUITS—2

In Chapter 3 the question of multiplication by analogue methods was mentioned, but only in the sense of multiplying a single voltage by some factor by altering the setting of a potentiometer (Fig. 3.1). To multiply two input voltages together there are various techniques available. One common method developed from Fig. 3.1 (a) is to apply the first voltage, V_1, to the element as shown and use the second voltage, V_2, automatically to position the wiper a distance from the bottom proportional to the value of V_2. To position the wiper an electric motor is used, and this is controlled from V_2 by a servomechanism as shown in

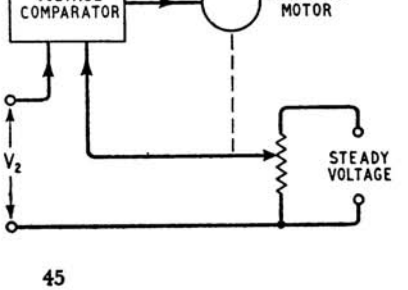


Fig. 4.1. Principle of the servo multiplier, based on the potentiometer multiplication method in Fig. 3.1 (a)

Fig. 4.1. The motor drives the wiper of a second potentiometer, across which a steady voltage is applied, until the voltage tapped off becomes equal to the input V_2. When this happens the motor is stopped. Because it has driven the wiper of the second potentiometer a distance proportional to V_2, it has also driven the first multiplying potentiometer the same distance.

This electro-mechanical arrangement is commonly known as a servo multiplier. It is comparatively simple and inexpensive to manufacture but has the disadvantage of being slow in operation for certain applications because of the mechanical inertia of the moving parts. Where high-speed multiplication is required, purely electronic techniques have to be used, although they may in fact prove rather more expensive in manufacture.

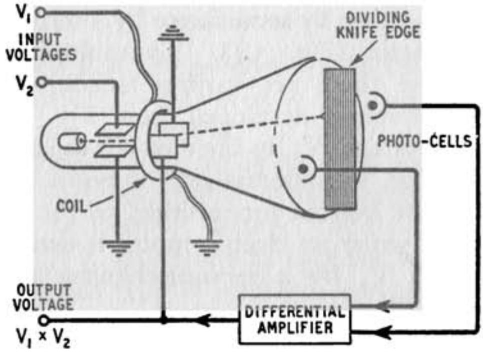


Fig. 4.2. Electronic multiplier based on a cathode-ray tube

As an example, an electronic multiplier based on a cathode-ray tube is shown in Fig. 4.2. There are the usual vertical and horizontal deflection plates in the neck of the tube and a coil is fitted round the tube near the horizontal deflection plates. One of the input voltages, V_1, is applied to this coil and produces a magnetic field along the axis of the tube of magnitude proportional to V_1. So long as the electron beam is undeflected, the electron beam current acts in the same direction as this magnetic field and does not interact with it. When, however, the second voltage, V_2, is applied to the vertical deflection plates, it imparts a vertical component of velocity to the electrons in the beam of magnitude proportional to V_2. This component of the beam current interacts with the axial magnetic field and produces a force on the electron beam, in the horizontal sense, of magnitude proportional to the product V_1V_2.

The tube face is divided by a vertically placed knife edge with a photocell on either side. This enables any horizontal deflection of the visible spot, and consequently of the beam, to be detected as a difference signal, which is amplified and applied to the horizontal deflection plates in a restoring sense.

To keep the spot on the knife edge, the restoring voltage has to counteract exactly the horizontal force applied to the beam. Since this horizontal force is proportional to V_1V_2, then the amplifier output is also proportional to the product of the two input voltages and is taken as the output voltage of the multiplier. Special tubes for this " crossed fields " type of multiplication have been developed in which the external optical arrangement is replaced by an internal electronic system, the beam being constrained to follow a slot in one of a pair of electrodes.

In another electronic multiplier the mark/space ratio of a rectangular wave is made to depend on one voltage, and its peak-to-peak amplitude on another. The detected and smoothed output then varies as the product of the inputs. This device has the advantage of being less expensive to make than the cathode-ray tube multiplier, and is comparable in accuracy with the servo multiplier without its inherent slowness of operation.

A multiplier can also be made from what are known as function generators arranged to give square-law responses.

If two such generators are arranged to evaluate $(A + B)^2$ and $(A - B)^2$, their difference, when scaled down by a factor of four, is equal to the product AB. Yet another type of circuit works on the principle of generating the logarithms of the numbers, adding them together and finally obtaining the antilog.

We have just mentioned the existence of function generators in analogue computers, and it will have been gathered that these are devices in which the output voltage is always a particular mathematical function (e.g., square, cosine, logarithm) of the input voltage. A simple electro-mechanical function generator is a potentiometer in which the element is specially shaped so that the output voltage (see Fig. 3.1) does not vary linearly with the movement of the wiper but according to some other law. Logarithmic and square-law potentiometers are quite well-known devices.

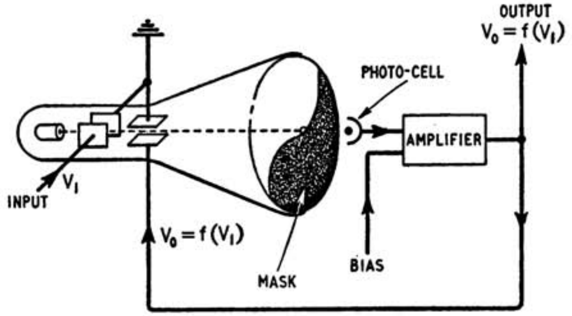
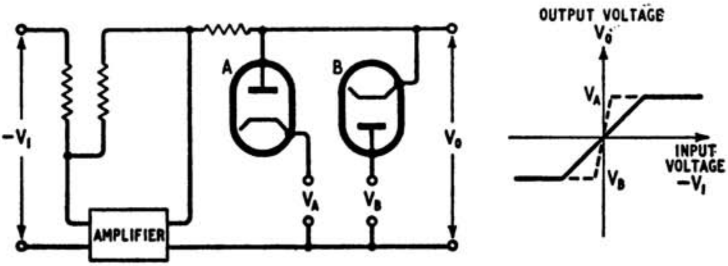


Fig. 4.3. Electronic function generator using a shaped mask on the face of a cathode-ray tube

An example of an all-electronic function generator is shown in Fig. 4.3. It consists of an opaque mask, having a profile representing the required function, placed on the screen of a cathode-ray tube. A photocell receives light from the fluorescent spot on the screen when this is not obscured by the mask, and the output of this photocell controls the voltage applied to the vertical deflection plates through an amplifier. The input voltage is applied to the horizontal deflection plates and causes the spot to move horizontally. A bias voltage is applied through the amplifier to the vertical deflection plates in such a sense as to cause the spot to rise until it reaches the edge of the mask. As the spot becomes unmasked the photocell input to the amplifier opposes the bias increasingly until the amplifier output attains a value at which the spot is prevented from rising further. The spot is thus forced to follow the mask profile, and the required function of the input voltage is generated at the output terminals of the amplifier.

When analogue computers are used to simulate mechanical systems it is sometimes necessary to have function generators



(a) (b)

Fig. 4.4. Two diodes arranged to give limiting effects on the output voltage from an amplifier are shown at (a), and the "saturation" characteristic obtained at (b)

which give "saturation" characteristics to the output voltage. In other words, the output varies linearly with the input over a certain range, but beyond that range is limited and does not change, as shown in Fig. 4.4 (b). This effect can be achieved by arrangements of diode valves with particular bias voltages. As an example, Fig. 4.4 (a) shows two diodes arranged to limit the output of an amplifier and (b) is the resulting output characteristic. Diode A conducts when the output voltage, V_0, exceeds the bias voltage

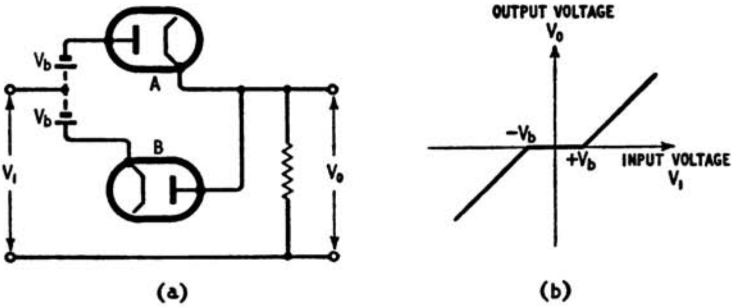


Fig. 4.5. Diode circuit for simulating a " dead zone " (a), and (b) the characteristic produced

V_A, so V_A corresponds to the maximum value of V_0 which can be passed. Conversely, diode B conducts when V_0 is less than V_B, so this voltage corresponds to the minimum value of V_0. The circuit gives a characteristic typical of a mechanical system with limits on position, rate or torque.

When a characteristic representing friction torque as a function of speed is required, a circuit similar to that in Fig. 4.4 (a) is used, but without the feedback connection to the amplifier input. The amplifier then drives its output to the limiting value for any slight departure of its input voltage from zero, in either the positive or negative direc-

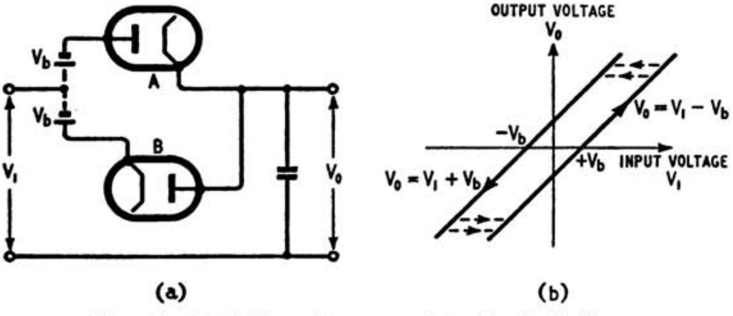


Fig. 4.6. Backlash can be represented by this circuit (a), as indicated by the characteristic (b)

tion. The resulting characteristic, which is of the desired form, is shown dotted in Fig. 4.4 (b).

A " dead zone " often exists in mechanical instruments and this can be represented in a computer by the arrangement of Fig. 4.5 (a). If the input signal V_1 is positive, it must exceed $+V_b$ for the diode A to conduct or, if negative, be less than $-V_b$ for the diode B to conduct. A positive or negative threshold of value V_b is thus introduced into the output V_0 as shown in Fig. 4.5 (b).

The inclusion of a capacitor at the output, as shown in Fig. 4.6 (a), somewhat modifies the behaviour of the circuit. The diode A now conducts when $V_1 - V_0 > V_b$, and B when $V_1 - V_0 < - V_b$. This results in the characteristic of Fig. 4.6 (b), which can represent, for example, the backlash in a system of gears.

The general idea can be extended to allow the approximation of almost any non-linear function by a number of straight-line sections, provided the slope of the characteristic remains of the same sign throughout. One example is shown in Fig. 4.7. At low input voltages, when the diodes are all biased off, the circuit acts as a simple potential divider, but as the input voltage increases the diodes become

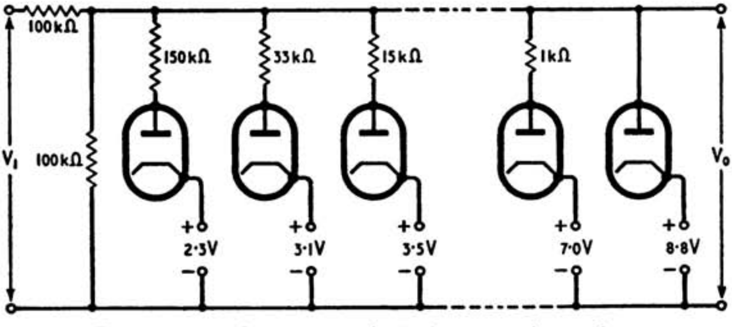


Fig. 4.7. Function generator circuit for compressing an input voltage range to a logarithmic scale

conducting one after the other, making the effective potential divider ratio progressively less. With the circuit values and bias voltages shown, the arrangement gives a logarithmic law over a wide range, and this particular circuit has been used as an attenuator for compressing voltages from 1 to 1,000 volts, so allowing the voltage values in both low- and high-level regions of the signal to be read on a meter to the same percentage accuracy.

Of course, the circuits described so far are not really intended for performing isolated calculations on fixed quantities, such as multiplying 2,469 by 335 and finding the square of the result. This kind of thing can be done much less expensively with pencil and paper, using perhaps a simple mechanical analogue device like the slide-rule. As explained in Chapter 2, the analogue computer really comes into its own when one has to do calculations on numbers or values which are varying with time—as may be necessary in the study of dynamic systems.

For this reason the process of integration is usually only required with respect to time, and the circuit in Fig. 3.5, based on the accumulation of charge in a capacitor, is therefore suitable for most purposes.

The circuits described do not find much use, either, as single units for performing isolated calculations, such as addition or multiplication, except perhaps where they are incorporated in, say, industrial process control systems. It is more usual to find groups of them arranged in larger circuits for solving more complex problems. Here, one of the most common arrangements is the differential analyser, mentioned in Chapters 1 and 2. As already explained, this is an analogue device which will solve differential equations when given some numerical value to start with. The differential equation, of course, is concerned with values that change, or more specifically rates-of-change, and in many dynamic systems under investigation this means rates-of-change with respect to time.

A very simple differential equation of this kind might take the form

$$\frac{\mathrm{d}x}{\mathrm{d}t} = y$$

where, say, x is distance, t is time and y is speed, so that $\mathrm{d}x/\mathrm{d}t$ represents the rate-of-change of distance with time o some object or vehicle (i.e., its speed). Usually in the analogue calculation one starts off with a value for y (so many miles per hour) and one wishes to find the value of x (the distance travelled) with respect to time. In practical terms, one has an input voltage proportional to a certain speed and one wishes to obtain an output voltage varying with time which gives the distance travelled at the various instants of time.

To get, so to speak, a " running total " of x with time one must obviously add up continuously all the small increments of x (represented by $\mathrm{d}x$ in the equation) as time advances. In other words, the mathematical process of integration with respect to time is called for. This process, applied to the present case, is expressed by the equation

$$x = \int y \, \mathrm{d}t$$

If y is represented by an input voltage V_1 and x by an output voltage V_0, it is clear that the process can be performed by the integrating circuit in Fig. 3.5 of Chapter 3, the action of which was given by a similar equation on p. 43.

Thus the simplest kind of differential analyser is just an integrator. A rather more complex arrangement is required, though, for solving differential equations which take the form

$$\frac{\mathrm{d}x}{\mathrm{d}t} + ax = y$$

where a is some known constant, and y is a known value available as an input voltage. In the first place it is necessary to rearrange the equation as follows:

$$\frac{dx}{dt} = y - ax$$

Now x can be obtained as before by integrating the value of dx/dt, but here this value is not just y but y with ax subtracted. It is therefore necessary to integrate a value, $y - ax$, of which the ax term is provided by the result of the integration (ignoring the constant a for the moment). This requires a kind of feedback process, which in practice is given by the type of arrangement shown in Fig. 4.8.

The circuits required are an adder (as in Fig. 3.3), an integrator (as in Fig. 3.5) a sign reverser (as described on p. 41) and a constant multiplier (using a modified value of input resistor to the adder, as explained on p. 36). A certain input voltage represents the known value y, while the output voltage obtained represents the wanted value x varying with time.

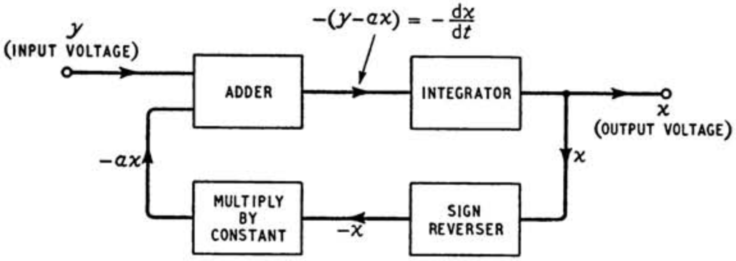


Fig. 4.8. Arrangement of basic analogue computing circuits for the solution of differential equations

To follow the operation it is necessary to assume that $- dx/dt$ exists at the output of the adder. When this is integrated x is produced at the output (the positive sign being obtained by making the integrator input voltage negative). The voltage representing x is fed back, reversed in sign to give $-x$, multiplied by the constant a in the adder to give $-ax$, and finally this is added to y to produce $y - ax$, but with negative polarity, $-(y - ax)$, because of the sign reversal in the adder. Referring back to the original equation, one sees that this, in fact, produces the actual $-\,\mathrm{d}x/\mathrm{d}t$ voltage which was assumed to exist at the input of the integrator.

Similar techniques are used for solving even more complex problems like second-order differential equations taking the general form

$$\frac{\mathrm{d}^2x}{\mathrm{d}t^2} + \frac{\mathrm{d}x}{\mathrm{d}t} + ax = y$$

or even third-order for fourth-order differential equations. With these, however, more than one feedback loop is required in the Fig. 4.8 type of circuit.

In talking about the *solution* of differential or any other equations it is important to remember that analogue computers are not devices for giving *general* solutions—which can be obtained much more simply by pencil-and-paper methods. They only solve particular cases for which numerical values are provided. It is always desirable to bear in mind that they are essentially calculating machines rather than algebraic problem solvers. Another point, already mentioned, is that analogue computers are not particularly useful when the input quantities have fixed values, for again the results can be obtained much more quickly on paper. It is when the input voltages are varying with time, perhaps in some quite complicated fashion, that the analogue computer really comes into its own.

There are three basic types of information which must be put into an analogue computer in order to do a calculation. By way of illustration, suppose that the simple equation:

$$y = 4x^3 + 7x^2 - 5x + 29$$

is to be solved by obtaining an output voltage representing y.

The first type of information is the form or pattern of the calculation, and this is put in by connecting together the appropriate computing circuits for adding, subtracting, cubing and squaring as required by the structure of the right-hand side of the equation. The second type of information consists of the constant terms (the 4, 7 and 5 constant multipliers and the 29 constant addend) which are put in by appropriate setting of controls (e.g., variable resistors) and voltages in the machine itself. The third type of information is the variable x, and this is fed in as the input voltage from outside. Thus the voltage for x can be varied continuously if necessary and the computer will give a continuously varying voltage for y at its output.

5

EQUIPMENT OF ANALOGUE COMPUTERS

THE previous two chapters have given some idea of the principles of electronic analogue computers, but what do these machines actually look like " in the metal "? Unfortunately the analogue computer has nothing particularly distinctive in its appearance to identify it, and for all that one can tell from a casual glance it might be any piece of electronic equipment. It has the same battleship-grey cabinet, with a control panel carrying switches, meters, variable controls and indicator lights, the same sort of cathode-ray tube screen, and the same supply of power from the mains. A look inside the cabinet reveals nothing more than conventional electronic circuitry.

Special-purpose machines are particularly unhelpful in this respect, and the equipment of one will be completely different from that of another. With general-purpose computers, however, one does begin to see some sort of uniformity in the design, though again this may be greatly influenced by the way in which they are used. There are actually two principal ways in which general-purpose machines can be utilized—as differential analysers or as simulators. In both situations the basic equipment remains much the same, but in the computer intended for simulator work there are likely to be a good many additional computing elements which are peculiar to its general field of application. A simulator intended for use in the design of automatic control systems will contain the sort of circuits described in the previous chapters, but also things like time delays, phase advancers, resonant circuits and other special devices to simulate control-system equipment.

Apart from these differences, however, practically all electronic analogue computers of the general-purpose type use d.c. amplifiers as their basic units. In fact, the smaller commercial models now becoming available are constructed almost entirely from these " building blocks ". If one looks inside the cabinet of such a computer, then, one can safely assume that most of the valves that one sees will belong to d.c. amplifiers. In a really large machine there might be over 500 of these amplifiers and in a really small one only about ten (p. 63, Plate 7[a]), but in the average sort of general-purpose computer (p. 18, Plate 2; p. 63, Plate 7[b] and p. 64, Plate 8) the number usually lies between 20 and 80.

As already explained, the d.c. amplifier has to be capable of amplifying steady input voltages, and voltages which are fluctuating at low frequencies below about 50 c/s, as well as perhaps higher frequencies in the audio range. This can be achieved by " chopping " the input voltage and amplifying the resultant pulsating waveform in an ordinary a.c. amplifier. The most common arrangement, however, is to use an amplifier in which the valve stages are directly coupled—that is, the anode of one valve is connected directly (or perhaps through a resistance) to the grid of the following valve.

In a.c. amplifiers, of course, there is usually a capacitor interposed. This allows the a.c. signal voltage to be passed but isolates the grid of the second valve from the positive h.t. voltage on the anode of the first valve. Thus, in the d.c. amplifier it is necessary to adopt some means of offsetting the positive voltages applied to the valve grids, and this is usually done by high-voltage negative biasing arrangements. Fig. 5.1 shows a simple directly-coupled d.c.

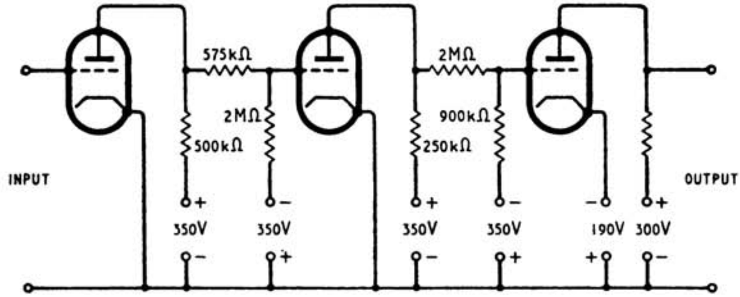


Fig. 5.1. Unbalanced d.c. amplifier with three stages to give overall sign reversal

amplifier in which the grids are biased in this way. It has a gain of about 50,000 and the output will vary over a range of about ± 150 V. Note that three stages are used to give the necessary sign reversal.

Another important requirement of the d.c. amplifier is that it should be free from " drift ". This is a name given to small irregularities in the operation of the valves which have the equivalent effect of introducing a small unwanted voltage at the input—which of course is amplified and produces a considerable unwanted voltage at the output. The ideal condition of operation is that with zero voltage applied to the input there should be zero voltage at the output. With " drift ", however, this spurious voltage is produced at the output which adds to the wanted amplified voltage and so affects the accuracy of the analogue calculation. Most practical d.c. amplifiers therefore have to include special means of neutralizing the effects of the drift on the output voltage. Fluctuations in the amplifier power supplies can also cause spurious effects in the output and again some means of compensating for them has to be adopted.

A practical type of d.c. analogue computing amplifier (slightly simplified) is shown in Fig. 5.2. Although it has six valves, there are in fact three amplifying stages to give

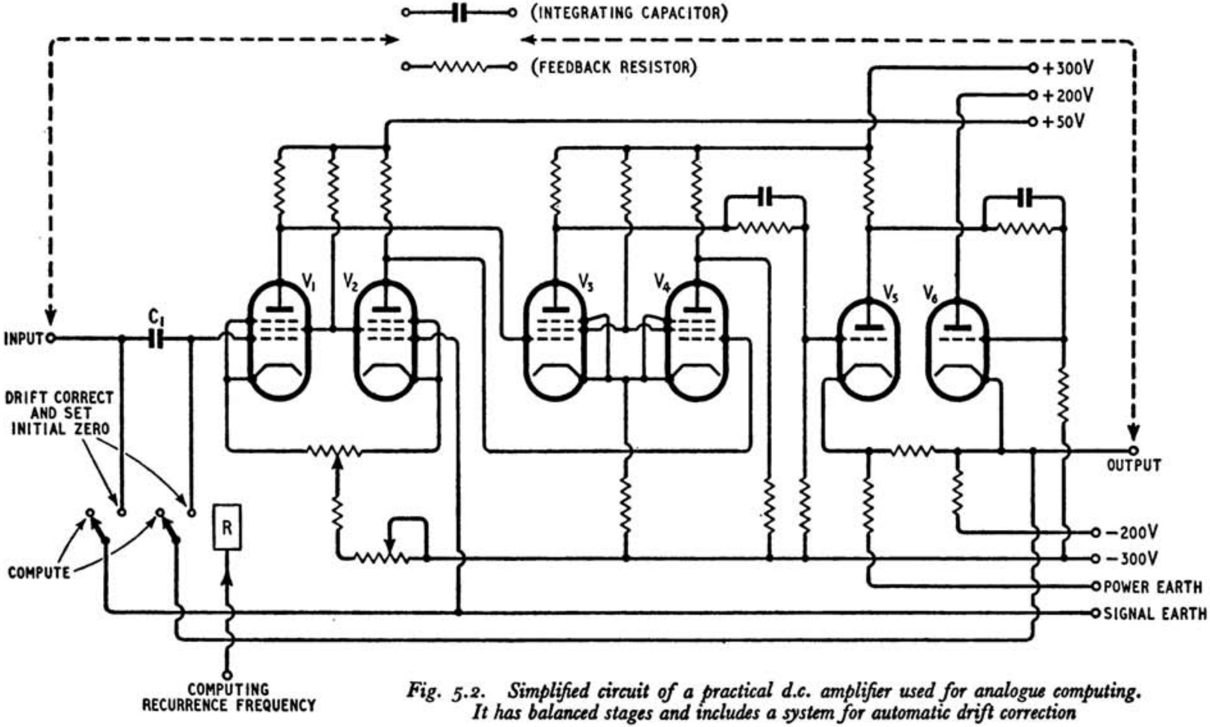


Fig. 5.2. Simplified circuit of a practical d.c. amplifier used for analogue computing. It has balanced stages and includes a system for automatic drift correction

the required change of polarity. The first stage consists of a balanced push-pull pair of valves, V_1 and V_2, coupled by a common cathode resistance. The push-pull output from this is fed to the second stage, which is another cathode-coupled pair V_3 and V_4. The third stage, fed from V_3, is the single valve V_5. The valve V_6 is merely a cathode-follower for giving a low-impedance output and preventing any loading on the output terminal from affecting the operation of V_5. It does not, of course, change the polarity of the output voltage from V_5.

Push-pull pairs of valves are used in the first two stages to balance out the effects of any power-supply or other common fluctuations. Because the output taken from a push-pull stage is of a differential nature (produced by one anode going positive while the other goes negative), any fluctuations in the h.t. or heater supplies affect both valves of a pair equally and so do not modify the differential output voltage. The potentiometer connected between the two cathodes in the first stage is for adjusting for zero output voltage during the setting-up procedure.

It will be noticed that there is a capacitor, C_1, in the input circuit to V_1. This is actually part of the drift-correction system of the amplifier. The system works on the principle of charging C_1 to the equivalent input drift voltage of the amplifier and then inserting it in series opposition with the input, when computing is in progress, so that the effect of the drift is cancelled.

When the relay contacts are in the right-hand " drift-correct " position, C_1 is connected between the output terminal and earth. It then becomes charged to any voltage caused by drift on the output terminal. In this condition, when the output terminal is connected back to the grid of V_1, it can be shown by a simple mathematical proof that the output voltage is equal to the equivalent input drift voltage. Thus, when the relay contacts are moved to the left-hand " compute " position, C_1 remains charged to the equivalent input drift voltage, but because of the polarity of the amplifier output this charge voltage opposes the equivalent input drift voltage and the effective drift voltage is reduced to zero.

Other methods of drift correction use auxiliary amplifiers which take the drift voltage as their input and produce a correction voltage at their output. This is applied as a bias voltage to the d.c. amplifier in such a way as to compensate for the drift error (for example to the grid of V_2 in Fig. 5.2). Usually the auxiliary amplifiers are a.c. types, which do not pass on slow d.c. fluctuations and therefore have no drift of their own. In order that they shall be able to amplify the drift voltage of the d.c. amplifier, then, this voltage must be modulated in some way. A common expedient is to use a relay vibrating at several hundred cycles per second as a " chopping " device to produce a square wave (whose amplitude, of course, corresponds to the drift voltage). The same relay can then also be used at the a.c. amplifier output to rectify the amplified square wave and so give the required steady correction voltage.

Looking at the switching arrangement at the input of the Fig. 5.2 d.c. amplifier, it will be realized that computing and drift correcting cannot be done at the same time. In fact they have to be done alternately, and in fairly rapid sequence, for the drift correction by this method is only effective for about a minute at a time (while C_1 holds its charge). This, then, implies a repetitive sort of operation in which the computing function of the amplifier (depending on the outside circuit) is performed over and over again with intervals between for drift correction.

A great deal of analogue computing is, in fact, done on this repetitive principle. The main reason is that the output voltage, being repeated in a regular sequence, becomes a waveform which can be displayed on the screen of a cathode-ray oscilloscope—an instrument which depends on repetitive coincident traces to give a stationary picture. And the

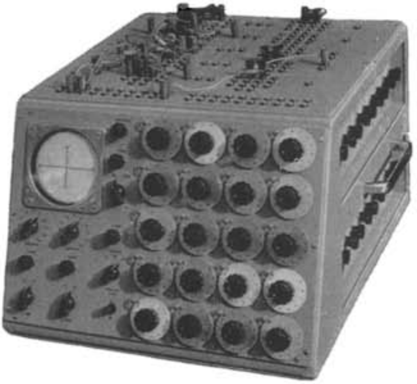


Plate 7 (a)—above. Small desk-type analogue computer made by Saunders-Roe, with patch panel on top and d.c. amplifiers plugged in from the right

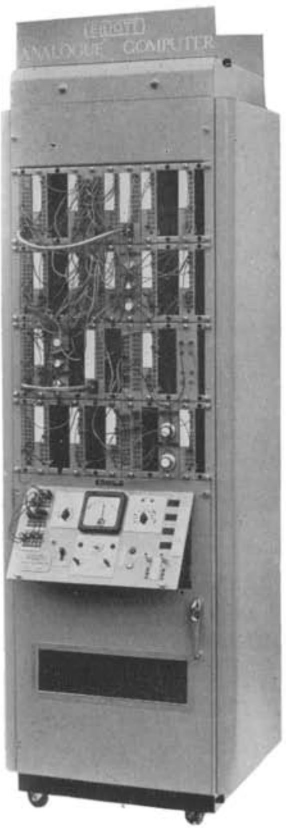


Plate 7(b)—above. General-purpose analogue computer designed by Elliott

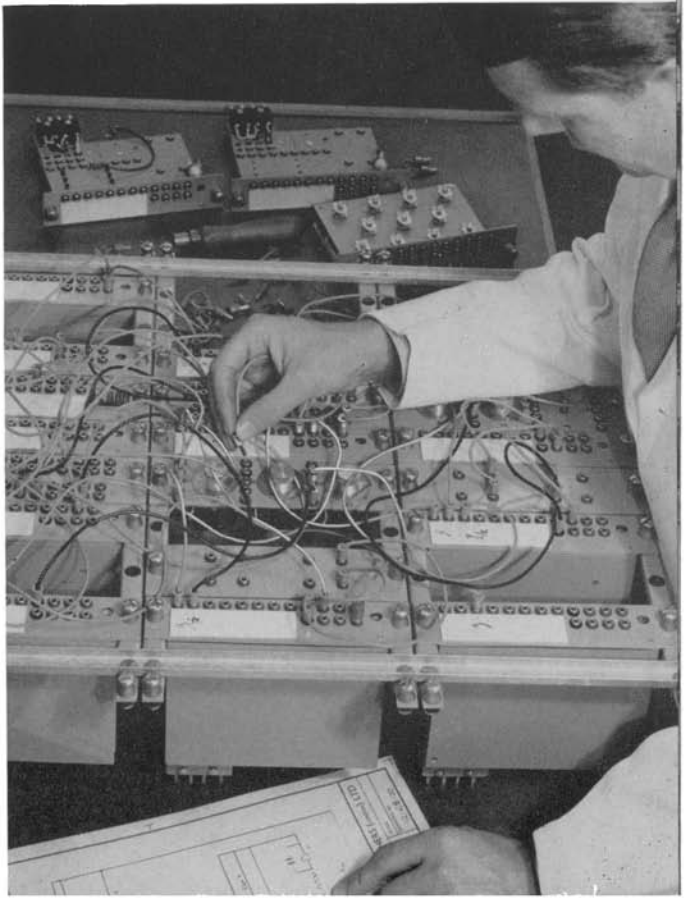


Plate 7 (c)—right. Setting-up a problem on the detachable patch panel of the Elliott machine

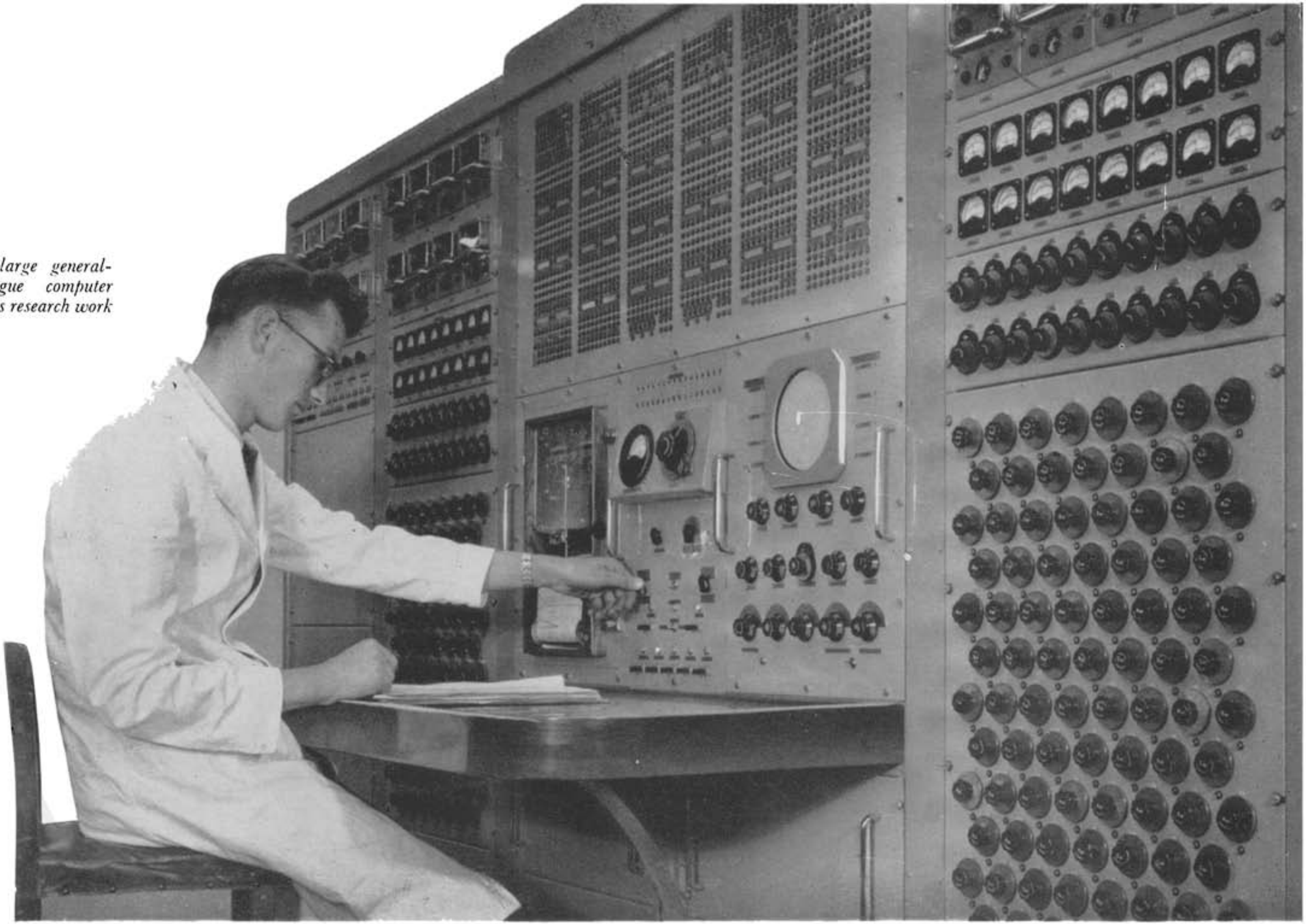


Plate 8. A large general-purpose analogue computer for aerodynamics research work

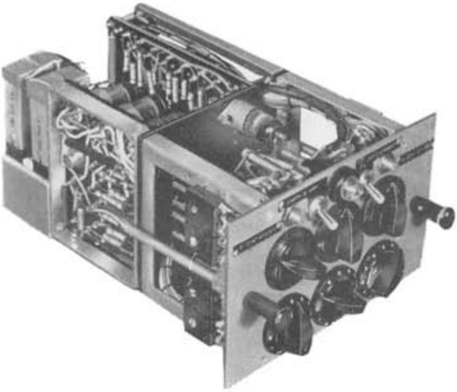


Plate 9 (a)—left. Detachable computing element, containing a d.c. amplifier, from the analogue machine in Plate 2, p. 18

Plate 9 (b)—below. Special camera arranged for recording solutions of problems displayed on the cathode-ray tube of the Plate 2, p. 18, analogue computer

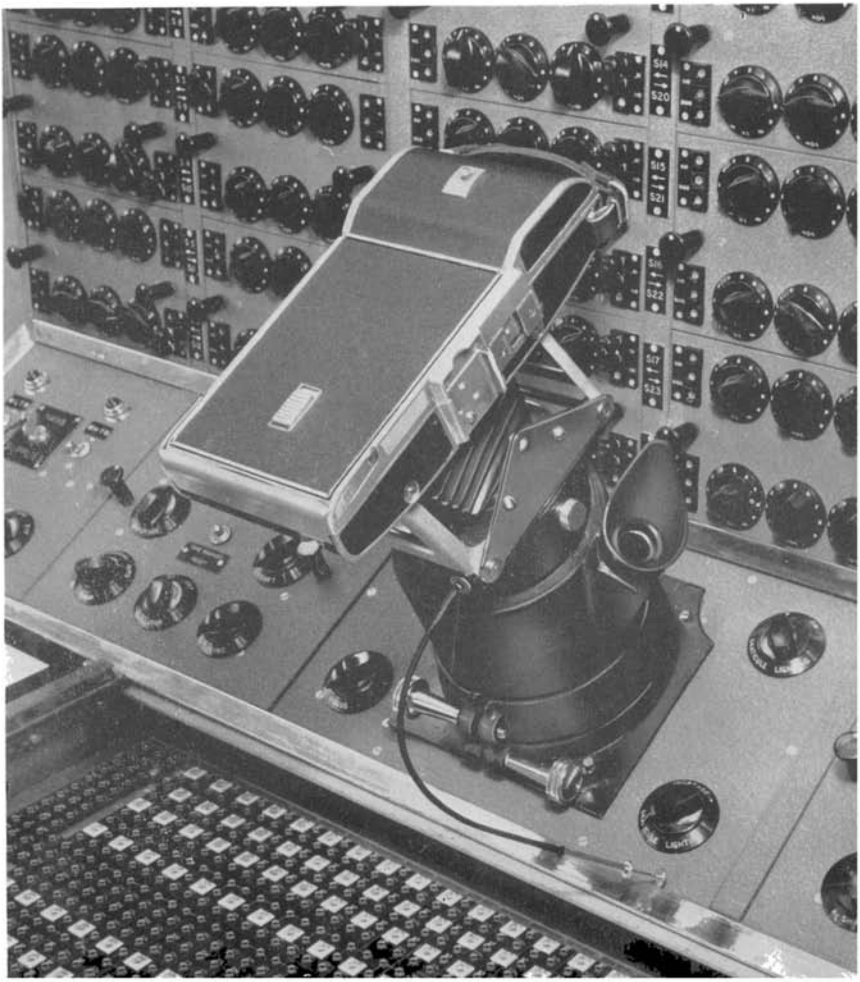
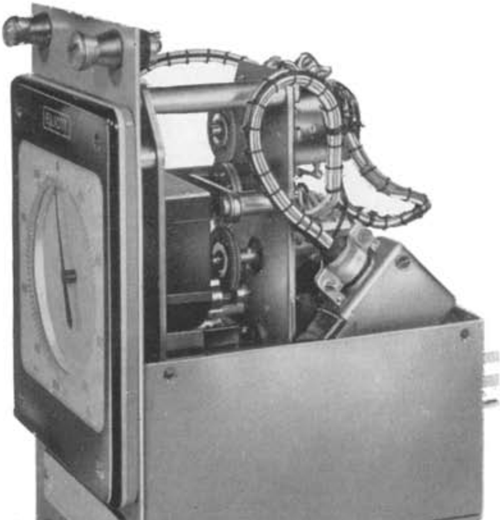


Plate 10 (a) — right. Electro-mechanical servo multiplier unit for an analogue computer



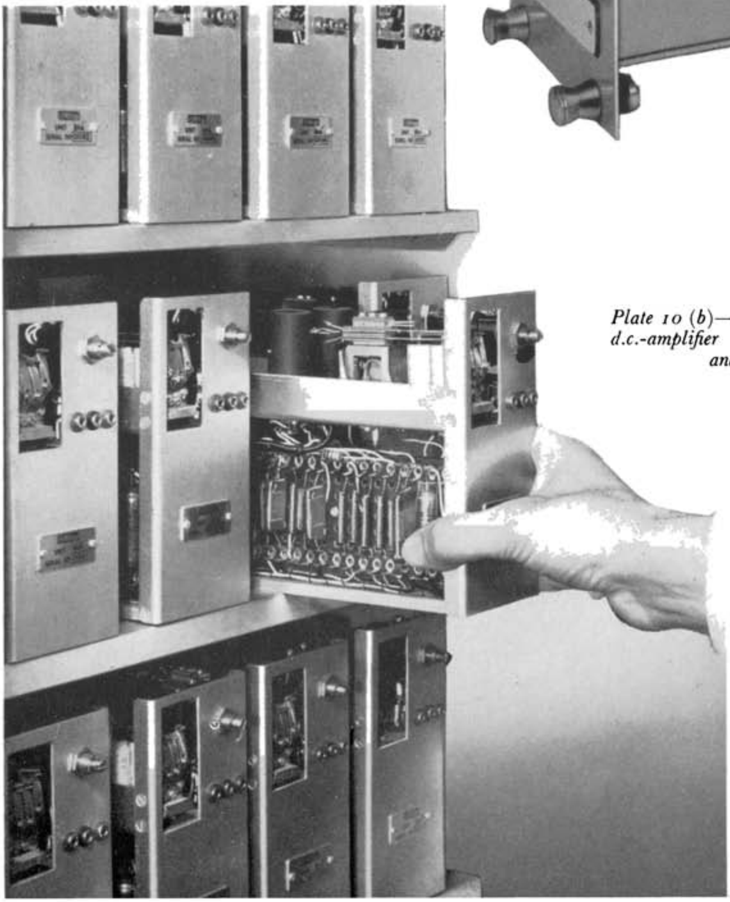


Plate 10 (b)—left. Standardized plug-in d.c.-amplifier units, as mounted in an analogue computer advantage of using an oscilloscope for displaying the output, as against a meter or other " single-shot " indicating device, is that one can adjust any variable elements, such as constant-multiplier resistances, while the computer is in operation and see the results of the adjustments immediately on the oscilloscope screen. With a needle indicator it would be necessary to do a series of separate " runs " and compare the results afterwards to see the effects of such adjustments.

Naturally the repetition rate has to be synchronized with the timebase of the oscilloscope to get a steady trace, and it has to be high enough to avoid flicker on the screen (quite often 50-c/s mains frequency is adopted). Moreover, the input voltage has to be repeated at the same rate and so also becomes a waveform instead of a steady voltage. The " computing recurrence frequency " applied to the relay in Fig. 5.2, then, arranges that the contacts are on " compute " when the oscilloscope timebase is performing its main sweep and that the timebase flyback occurs in the interval when the relay contacts are on " drift correct ".

In some modern analogue computers the d.c. amplifiers are built as small standardized units which can be plugged in or withdrawn as required (see p. 65, Plate 9[a]; and p. 66, Plate 10[b]).

This makes for a very flexible system of " functional units ", rather on the Meccano principle, from which any kind of circuit configuration can be built up. It also helps a great deal with fault-finding and general servicing. In Plate 10 (b) each of the amplifier units has a cathode-follower output and is drift corrected by means of an auxiliary circuit containing a " chopper " relay and an a.c. amplifier. The gain at zero frequency is about 5×10^6 and at 50 c/s about 11×10^3 times. The maximum output voltage is 100 V.

We have already seen from Chapters 3 and 4 that the d.c. amplifiers perform analogue computations in conjunction with resistors and capacitors and are connected up together in various ways. In a general-purpose machine, then, it is essential to have a convenient and flexible method of making these connections, so that the same basic units can be arranged in different computing circuits without too much trouble. This is generally done by a system rather like a jackboard in a telephone exchange.

It is commonly known as a " patch panel " and it can be seen in the upper half of Plate 7 (b), p. 63, on the top of the small machine in Plate 7 (a), p. 63, above the cathode-ray tube screen in Plate 8, p. 64, and at the bottom of Plate 9 (b), p. 65. The input and output terminals of all the d.c. amplifiers, the terminals of the resistors and capacitors, and of any other elements, are all taken to sockets on the panel. Connections between them are then made by flexible cords with plugs on each end (see p. 63, Plate 7 [c]).

This system, although rather untidy in appearance, is very flexible and can be modified in various ways. For example, the resistors and capacitors can be inside the cabinet, with connections brought out to the sockets, or they can be separate items plugged into the patch panel externally. Then again, in one particular machine (Plate 7 [c]) the complete patch panel is formed by a number of smaller panels, each of which belongs to a detachable unit containing resistors, capacitors or combinations of both. In some of these small units all the connections are brought out to the sockets on the front panel, while others are arranged so that when the unit is put into the patch panel it also plugs into a d.c. amplifier at the rear. This again is based on the Meccano principle and makes for great flexibility in building up different computing circuits.

Some analogue computers use banks of rotary switches for selecting the resistors and capacitors to be connected to the d.c. amplifiers and reserve the patch panel for linking together the computing units so formed. An example of this is the computer in Plate 2. Here the impressivelooking vertical array of control knobs (seen also in Plate 9 [b]) is actually made up of the front panels of a number of small computing units, which are inserted into the main rack and secured by quick-release fasteners. These small chassis are mechanically interchangeable and can be arranged in various ways. Electrical connections are made automatically by plugs which engage with sockets on the main frame.

A typical one of these computing units (similar to that in Plate 9 [a]) contains a d.c. amplifier and a number of resistors and capacitors. On the small front panel one of the rotary switches selects particular networks of resistors and capacitors (of the kind shown in Fig. 3.5 in Chapter 3) for connection to the amplifier. Seven configurations are available altogether, and these, in conjunction with the amplifier, form what are known as " transfer functions " for representing individual elements of a physical system when the computer is used as a simulator. Other rotary switches on the panel select different values of resistors and capacitors. A pre-set potentiometer control is used to adjust for zero output voltage during the initial setting-up process, while a toggle switch puts the automatic drift correction in and out of operation as required.

The unit shown in Plate 9 (a) is actually a computing element for producing the " limiting " and " dead-zone " conditions shown in Figs. 4.4 and 4.5 of Chapter 4. It contains a d.c. amplifier basically the same as that described above, another amplifier, and a pair of diodes arranged as shown in the Chapter 4 figures.

An important control on the front of this unit is a rotary switch for selecting the type of discontinuous function required. In one position this connects the two amplifiers in series to give straightforward linear amplification, in another it gives the " limiting " characteristic and in the third position the " dead zone " characteristic. Other controls are provided for setting the voltages which determine the actual extent of the "limiting" and of the "dead zone".

Not far from the controls on the front panel of the computer one will find some kind of indicating device for displaying the result of the computation. It might be just a meter of the pointer type, a pen recorder, a cathode-ray tube or possibly all three. Where the solution of a problem expressed in analogue form is simply a steady voltage the pointer instrument is quite sufficient. In most analogue computer work, however, the result of the calculation is an analogue voltage varying with time, and here, of course, the pen recorder or cathode-ray tube is called for. The machines shown in Plates 2 and 8 have both of these. In Plate 8 the chart of the pen recorder is just to the left of the operator's left hand, while in Plate 2 the recorder is housed in a drawer in the right-hand pedestal of the desk.

The cathode-ray tube is, of course, provided with a timebase and deflection amplifier to form the oscilloscope mentioned earlier. Sometimes these electronic circuits are built into the computer while sometimes the oscilloscope is an entirely separate instrument (as with the Plate 7 [b] machine). A typical sort of oscilloscope trace is shown in Fig. 5.3, and a similar kind of thing could be produced by a pen recorder.

One disadvantage of the cathode-ray tube, of course, is that it does not give a permanent record of the repetitive

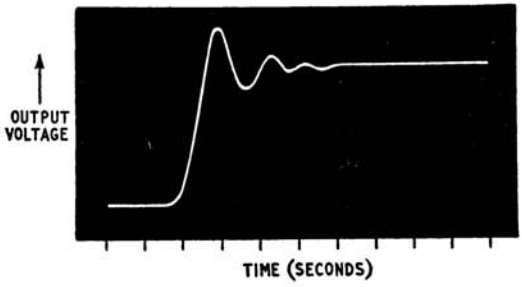


Fig. 5.3. The solution of a problem by an analogue computer often appears in this form, as a graph of voltage varying with time

solutions, and to overcome this some analogue computers are equipped with recording cameras. Thus in Plate 9 (b), p. 65, the camera is designed so that two exposures can be made on each film frame, giving a total of sixteen per roll of film. As soon as the exposure is made the film is moved forward and this sets in motion an automatic developing process inside the camera which produces a fixed positive print in the space of about a minute.

All analogue computations have to start with certain values for the voltages which represent the variables, and of course in many cases these initial values are just zero. If the calculation demands that an adder, integrator or some other computing element should have a constant value at its output terminal before the input voltage is applied, then a steady voltage representing that initial value must be applied from some source in the computer. If the initial value has to be zero, a simple earthing of the appropriate point in the circuit is all that is necessary. In Fig. 5.2, for example, when the left-hand relay contact is at the " drift correct " position, the external feedback resistor or integrating capacitor is disconnected from its normal input and connected to earth. This ensures that the voltage on the output terminal is zero (taking into account the drift correction) when the relay contacts move over to the " compute " position and the calculation begins.

This initial zero setting takes place repetitively, of course, when the computer is in repetitive operation. With the integrating capacitor, the plates are discharged at the end of each computing cycle by this arrangement. Thus there is no chance of any residual charge from one cycle being added to that of the next cycle and so introducing an error into the calculation.

When the initial value at the output terminal has to be something other than zero, different switching arrangements are necessary. In, say, an adding circuit, the input terminal has to be connected through an extra input resistor to a steady voltage of the required value at the beginning of the " compute " period. This value then appears at the output terminal and any other input voltages are automatically added together " on top of it ". In an integrator, the capacitor is connected to a source of voltage and charged to the required initial value before the computing cycle begins. The small computing unit in Plate 9 (a) has two control knobs for selecting the voltages required to represent initial conditions.

Sources of voltage are also required to represent the values of the variables involved in a calculation. In many computations these can be just steady voltages, which are applied by switches to the appropriate points. With repetitive operation the steady voltage is regularly interrupted to form a square wave. This can be produced either by " chopping " a d.c. source with vibrating relay contacts or by an electronic generator, but in either case the wave has to be synchronized with the computing repetition frequency.

Many calculations in which the computer is used as a simulator require an input variable in the form of a sudden rise of voltage or " step function ", for example in simulating the response of a mechanical system to the sudden application of a certain force (Fig. 5.3 shows such a response). This again can be produced by switching on a steady voltage, or, in repetitive operation, by the sharp front edges in a train of square waves. An input variable which is linearly rising in value is called a " ramp function " and in repetitive operation takes the form of a sawtooth wave.

A fairly elaborate system is sometimes required for synchronizing the operation of the various switching relays with the timebase of the cathode-ray tube display. In the Plate 2 computer, for example, a central timing generator is used as a synchronizing reference.

A sine-wave oscillator forms the basis of the timing generator, and its frequency can be switched to 300 c/s, 100 c/s, 30 c/s or 10 c/s. Pulses are derived from the sine waves and, after frequency division, are used for triggering the timebase of the cathode-ray tube display, for suppressing the cathode-ray tube flyback trace, for controlling the step-function relay (as mentioned above) and for operating the drift-correction and initial-condition relays. Each sawtooth in the timebase waveform occurs in the first of two equal intervals, and it is during the second " non-computing " interval that the drift-correction and initial-condition relays come into operation. The timebase sweep durations actually made available are 1/30th second, 1/10th second, 1/3rd second and 1 second. A double-beam tube is employed for the cathode-ray tube display and the second trace is generally used for presenting a series of time-marker pulses—which are also obtained from the timing generator.

Probably somewhere near the bottom of the computer cabinet will be found the power supplies for the electronic circuitry. In the Plate 2 machine, for example, six power supply units are fitted into racks in the left-hand pedestal of the desk and electrical connections are made by an automatic plug-in system. Some of the valves in such power supplies will be rectifiers, while others will be stabilizers for the h.t. supplies of the d.c. amplifiers. It is, of course, particularly important that there should be no variations in the h.t. voltage of these units, otherwise the output voltages would be affected and inaccuracies would be introduced into the computations. Balanced d.c. amplifiers avoid this trouble to a great extent, but with the unbalanced type shown in Fig. 5.1 stabilization is really essential.

Another good reason for stabilizing the h.t. supplies is to eliminate any ripple which may be present in spite of the smoothing circuits, for this again could introduce inaccuracies. Perhaps the most important reason of all, however, is to keep the internal impedance of the h.t. supply as low as possible. This is necessary to avoid any unwanted coupling between d.c.-amplifier elements which could be produced by a common supply. With such a coupling it is quite likely that any changes in the h.t. currents drawn by some of the amplifiers could affect the h.t. voltage and so influence the output voltages of other amplifiers.

In general, analogue computing equipments are nowadays tending towards greater simplicity and cheapness to cater for the growing market of educational establishments and small firms who cannot afford large and expensive general-purpose computers. Manufacturers are beginning to realize, in fact, that there is no point in trying to produce a truly general-purpose machine since most users' requirements are specialized or restricted to a certain field. The tendency, therefore, has been to manufacture either small desk-type machines of limited capabilities or separate computing elements from which complete equipments can be assembled on the " build-it-yourself " principle.

6

APPLICATIONS OF ANALOGUE COMPUTERS

SOME idea has already been given of the type of work for which analogue computers are built, and one can see that it is largely engineering research and design. Quite often in this field an analogue computer is constructed specially to solve a particular type of problem. In the majority of cases, however, the problem to be solved does not justify the building of a large special-purpose computer and it has to be taken to a general-purpose machine capable of handling a wide variety of engineering calculations. Whatever machine is used, however, the analogue computer has certain basic characteristics which make it particularly suitable for this kind of work.

Probably the most useful of these characteristics is the fact that one can vary the input data on the computer control knobs as the machine is working and see the result immediately in the computed answer. For example, suppose the computer is doing a calculation in aerodynamics to find the displacement of an aircraft fuselage in a certain direction resulting from a given deflection of a control surface. It may be necessary to see how the aircraft displacement *varies* with different deflections of the control surface. This is done in the analogue computer simply by altering the voltage representing the deflection by means of a potentiometer and watching the effect on the indicating system which gives the displacement solution. But to get the same information from a digital computer it would be necessary to perform a separate calculation for each movement of the control-surface deflection and then finally assemble all the answers in tabular or graphical form—a business which might take several hours in a complex problem.

This facility of being able to vary a quantity in the analogue computer corresponding to a physical variable in the mechanism being studied is particularly useful in a special class of analogue machines known as simulators. These have already been mentioned in previous chapters. The simulator is so called because it is intentionally constructed as a " model " of the actual mechanism under investigation, with all the variable quantities in the computing apparatus having definite counterparts in this mechanism. Moreover, it is usually (though not always) arranged to work in the same time scale as the actual mechanism, so that if a particular operation takes 100 milliseconds in the real thing the corresponding operation in the simulator also takes 100 milliseconds.

A good many ordinary analogue computers of the differential analyser type happen to work in this fashion simply because the analogue principle is to some extent a simulating principle; but they are not purposely designed as simulators, and the fact that some of their inner processes may be a " model " of the real thing is of no value in the calculation. From the user's point of view the computer can be regarded as a " black box " receiving problems at the input and giving solutions at the output. The apparatus inside the box is of no concern as long as it produces the right answers.

The simulator, however, is an analogue computer in which the various computing elements (adders, integrators, multipliers etc.) have been chosen and connected together in a special way to represent a particular type of mechanism. (And the word " mechanism " is used here in the broadest possible sense. Although generally it refers to something like an automatic-pilot control system in an aircraft, it can equally well apply to, say, the behaviour of an economic system.) There is a part-by-part correspondence between the simulator and the real thing, and each computing element solves the mathematical equation which expresses the physical behaviour of its counterpart in the mechanism under investigation. This feature is particularly valuable in research and design work, because it enables the engineer to get the " feel " of what he is doing—the reality behind the sets of mathematical expressions. Another quite important advantage of the simulator, compared with orthodox calculating machines, is that the engineer can insert an actual part from the real mechanism into the circuitry (for example a servo motor from an aircraft control system) and so obtain more realistic results than with the equivalent computing element. It is for this purpose, in fact, that simulators are generally designed to work in the same time scale as the systems they are representing.

The fact that the simulator elements are " models " of the corresponding parts of the real mechanism does not mean that they are physically analogous, but just that they behave in the same way. In other words, the voltages at the input and output terminals of a computer element have the same relationship to each other as the variable quantities of the input and output of the corresponding real apparatus. It is a matter of function rather than structure. This relationship of the output voltage to the input voltage of a computer element (actually their ratio) is, in fact, the " transfer function " referred to in Chapter 5. Thus, in simulating a mechanical system it is not necessary to write down all the equations describing the behaviour of the system and solve them by straightforward analogue computation. One merely has to select computing elements with the same transfer functions as the mechanical elements and connect them together as the mechanical system is connected.

Some examples from aerodynamics and aircraft control systems have been mentioned above, and it is in fact true

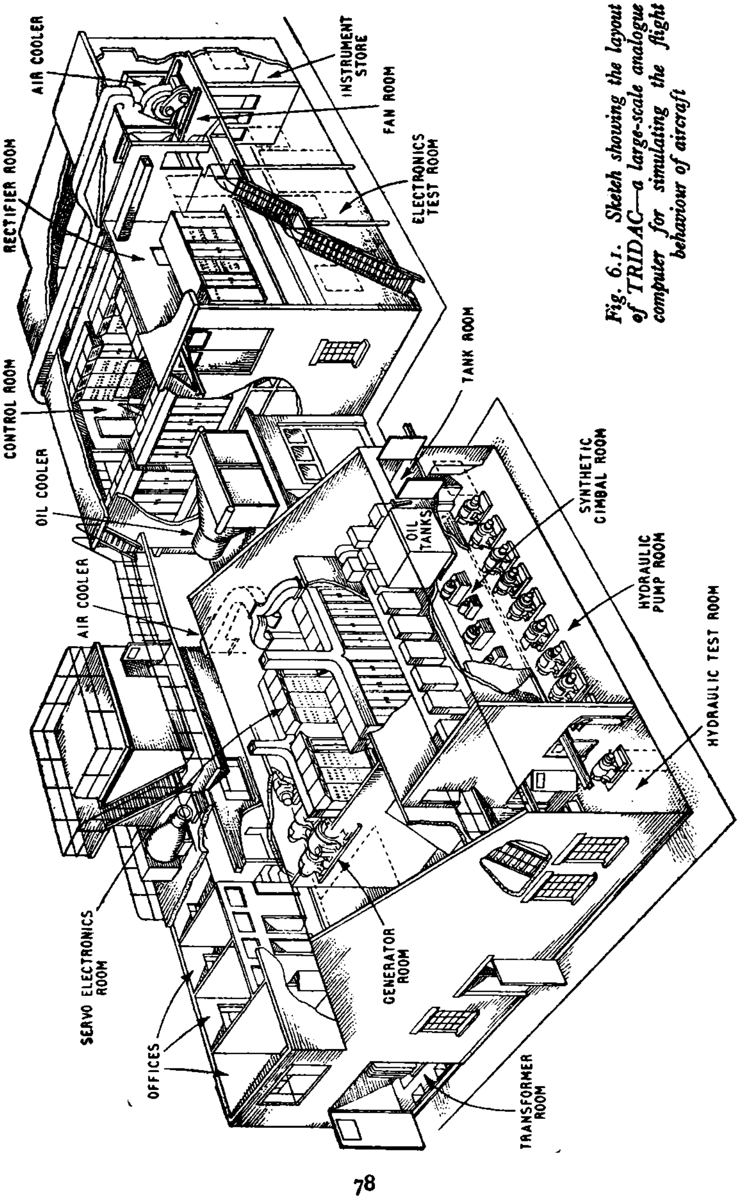


Fig. 6.1. Sketch showing the layout of TRIDAC—a large-scale analogue computer for simulating the flight behaviour of aircraft

that aviation development work has so far provided the greatest field of application for analogue computers and particularly simulators. Amongst the orthodox calculating machines, the differential analyser figures very largely here as a means of solving " flutter " and vibration problems in airframe design.

The " flutter " is actually a spurious mechanical effect which occurs in flight when several control surfaces have vibrations which interact to produce a self-supporting oscillation. The analogue computer is required to find the extent of this oscillation in various designs, or stages of design, so that the flutter can finally be eliminated. This means that the computer must be capable of simulating a number of vibrating mechanical systems. In each of these the relationship of such quantities as mass, acceleration, velocity, stiffness, damping and displacement is expressed by a second-order differential equation, and the computer circuit is similar to the differential analyser of Fig. 4.8 but with another integrator and feedback connection.

While the orthodox analogue computers are used mostly for solving particular aerodynamic problems, a number of simulators have been built to represent complete aircraft in flight—for example, guided missiles under radio control. One of the most ambitious of these is a machine known as TRIDAC at the Royal Aircraft Establishment, Farnborough. The name TRIDAC is derived from " three-dimensional analogue computer ", which means that it is designed for solving problems of flight in three dimensions. Altogether it contains over 600 d.c. amplifiers and a good deal of electro-mechanical computing apparatus—an assemblage which requires a large building to house it, complete with offices, stores, test rooms and its own power station. Fig. 6.1 gives some idea of how the whole complicated machine is arranged.

The type of flight behaviour which TRIDAC is mainly intended to simulate is that of a guided missile automatically

" homing " on to a moving target, with the aid of a radar system, either on the ground or in the missile. The object of this is to study how the missile behaves when chasing its target, which may be taking violent evasive action, and to see if it does indeed make a successful interception. To carry out whole series of such tests with real missiles and targets would be an extremely expensive and time-consuming business, but the cost of doing the equivalent runs on TRIDAC is insignificant by comparison. In order to imitate these conditions, then, the machine has to have computing elements which will represent the behaviour of the target and all the various processes of the missile control apparatus including the radar system. The resulting manoeuvres of both target and missile are traced automatically by moving pens on two large plotting tables (p. 83, Plate 11 [a]), one giving a " plan view " of what is happening relative to the points of the compass, and the other continuously indicating the missile and target height —that is, their relative positions in the third dimension.

Deriving the necessary signals to move the pens involves a long sequence of operations. Voltages representing outputs from the missile's automatic-pilot system are fed to computing elements which correspond to the motors driving the aerodynamic control surfaces. Further voltages equivalent to the control-surface deflections are then derived, and from them the resultant angular and linear accelerations of the missile in its own three axes are computed. Integrating circuits turn the acceleration values into angular and linear velocities, which are next converted into speeds relative to the ground. Three more integrators then derive the actual distances moved by the missile in three dimensions, and it is the output voltages of these which are used to control the " missile " pens on the plotting tables.

At the same time a " target-motion generator " produces voltages, according to a prearranged programme, which represent the velocities of the target in three axes relative to the ground. These, like the missile velocities, are integrated to give displacements, and the displacement voltages are used to control the " target " pens on the plotting tables. If the missile is supposed to be controlled by a radar and radio installation on the ground, the control signals fed back to the simulated automatic pilot are obtained from the three-dimensional displacement information (on both missile and target) given by the plotting tables, which is relative to the ground. But if airborne radar is assumed, the control signal must obviously be computed in a different way, for the distance information is then relative to the missile itself.

The operator of the machine can also observe what is happening on various meters and cathode-ray tubes (see p. 83, Plate 11 [b]) and he can stop the action at any time he wishes. By simply adjusting the controls he can alter the aerodynamic characteristics of the missile, while to obtain effects like gusts of wind he has only to add suitable impulses to the voltages representing the missile velocities. Almost everything is variable, in fact, so that the operator can experiment with the model to find the best conditions for intercepting the target and so obtain practical results for use in the design of the actual missile. Incidentally, the cost of building a simulator of this size (apart from development work) is about £250,000.

Whereas machines like TRIDAC are built for research and design work, there is another class of flight simulators intended for the rather different purpose of training air-crews. Again the main idea is to save expense. Normally it costs over £6,000 to train a crew for a large airliner—that is, about 21 hours' flying time at £300 per hour—but the cost of running a flight simulator is only £5 per hour and it enables the actual flying time in the real aircraft to be reduced to something like five hours. Overall expenditure can be cut by about 60 per cent.

The basic principle of the training simulator is, of course, much the same as the well-known Link trainer—that is, it provides instrument readings in response to the pilot's control movements—but the facilities needed to represent a modern airliner are of such immense complexity by comparison that the original idea has been developed almost out of recognition. First of all an exact replica of the aircraft's cockpit has to be built, complete with controls and instruments for the captain, co-pilot and flight engineer. Plate 12 (a), p. 84, shows a typical installation built by Redifon for B.O.A.C. All the controls are fitted with potentiometers which supply voltages, depending on their position, to electro-mechanical servo mechanisms grouped in three computer units dealing respectively with the flight parameters and with the inner and outer pairs of engines. The servos and their associated amplifiers are given characteristics, corresponding to the performance of the part of the aircraft which they simulate, which are incorporated in the form of potentiometer cards of specially calculated contours. The servo outputs are integrated in such a way that all interrelated aspects of flight performance are varied simultaneously and in the correct ratio. Appropriate meter readings are then passed back to the cockpit instruments.

For example, loss of power in one " engine " puts a calculated yaw on the aircraft which affects the synthetic compass reading and also provides a physical reaction which the pilot can feel in the controls. Changes of airspeed follow manipulation of the engine throttles and these are associated with appropriate changes in the " feel " of the flying controls. Engine noises (one loudspeaker for each engine) are accurately related in intensity to the brake horse-power readings and in frequency to the revolutions per minute shown on the indicator, so that the engineer can check synchronization. Aerodynamic noise is supplied by a unit coupled to the airspeed indicator, and there is a replica of tyre noise which is injected at the exact moment to indicate touch-down.

The instructor, who sits behind the flying crew, is provided with a " trouble panel " from which he can simulate no fewer

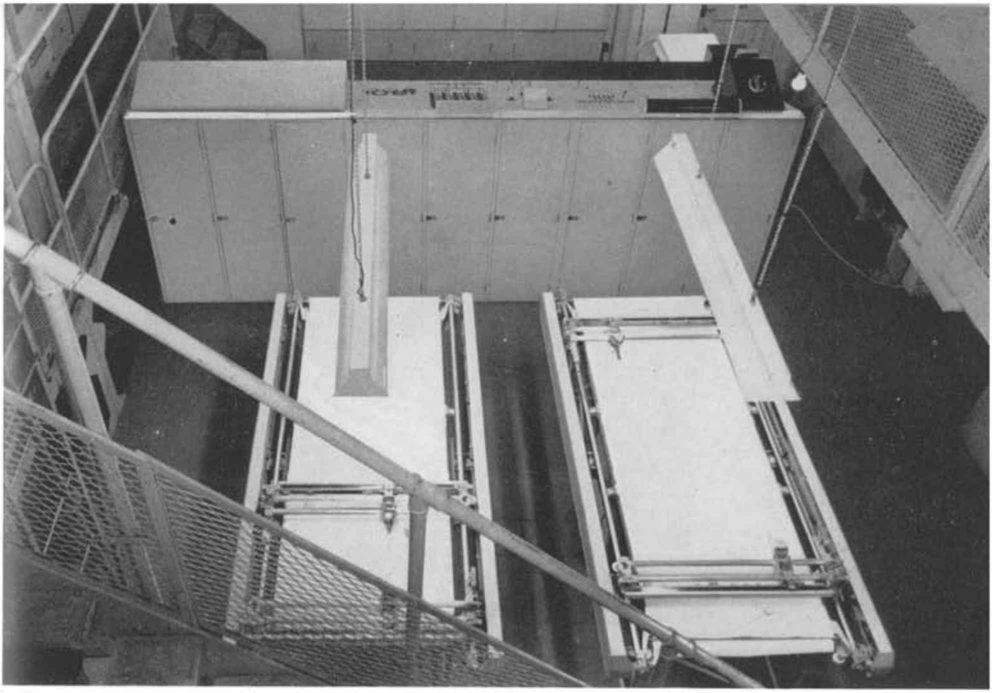
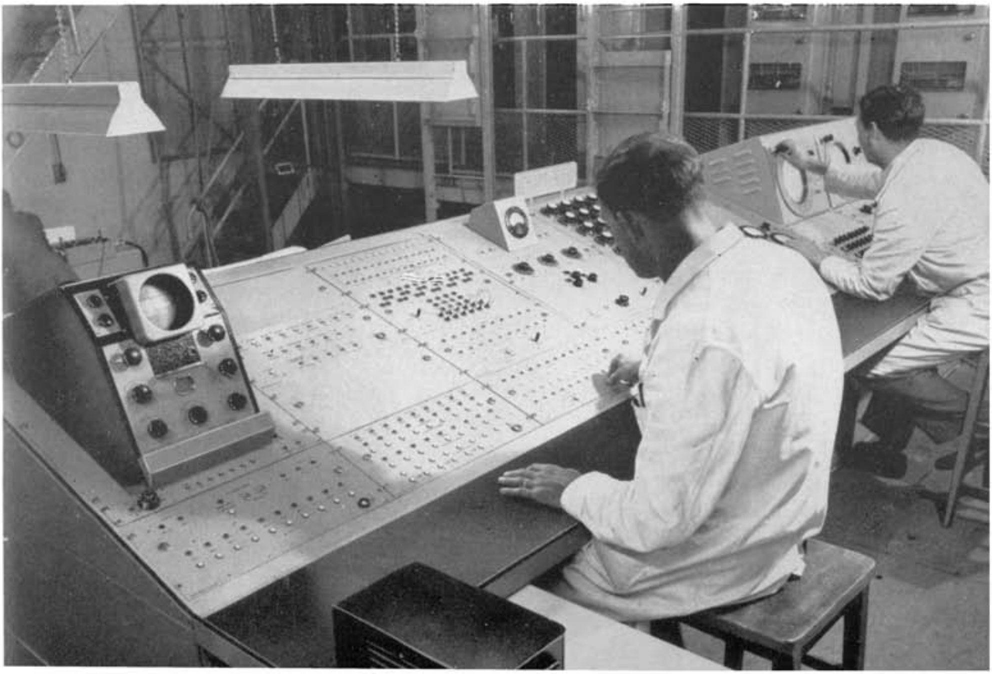


Plate 11 (a). The two plotting tables of TRIDAC, for recording the flight paths of simulated aircraft or missiles in three dimensions

Plate 11 (b). Control desk of TRIDAC, with cathode-ray tubes for displaying simulated flight behaviour on the right



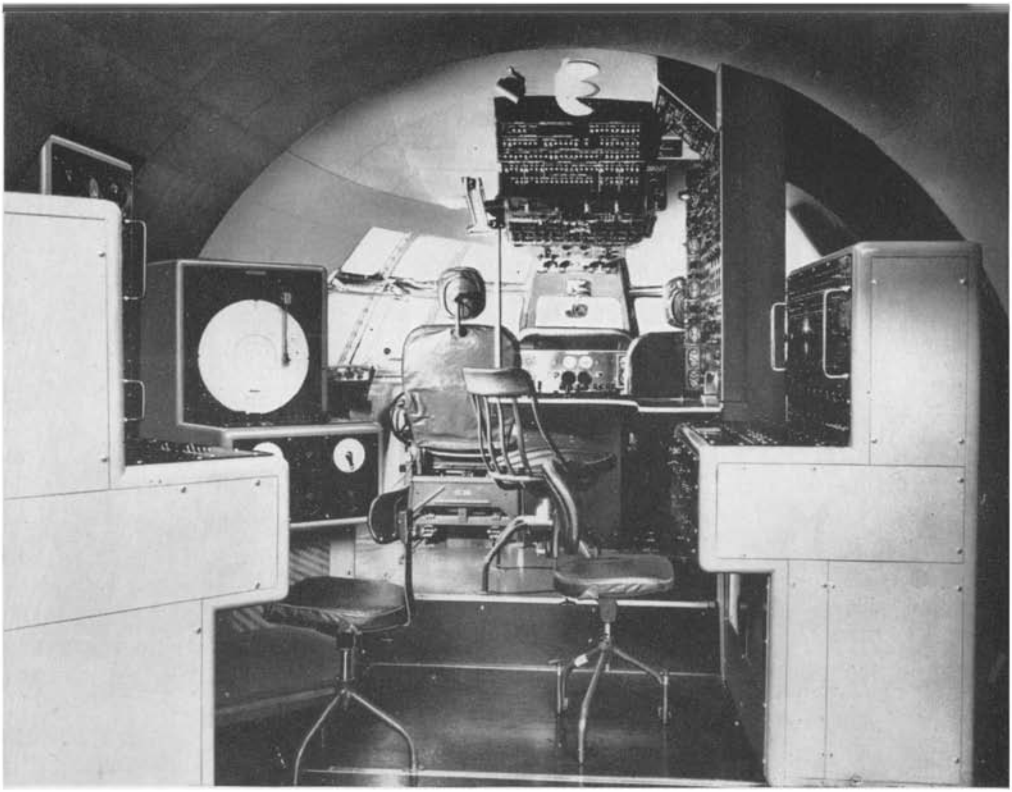
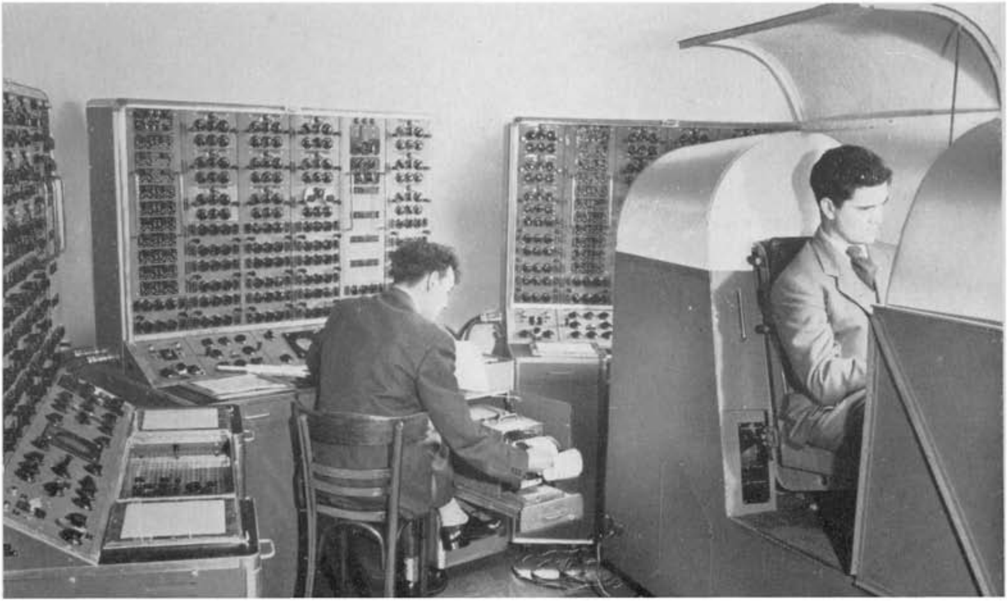


Plate 12 (a). Interior view of the " pilot's cabin " of a flight simulator built by Redifon for training aircrews

Plate 12 (b). Another flight simulator system, using conventional analogue computers of the type shown in Plate 2



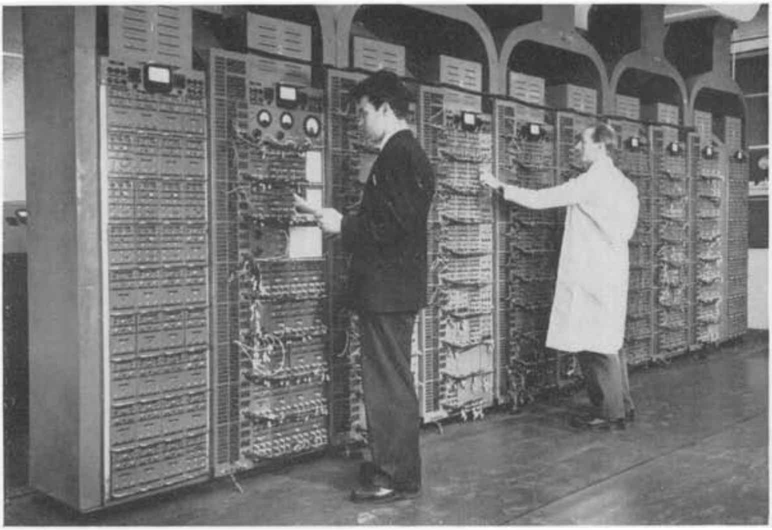
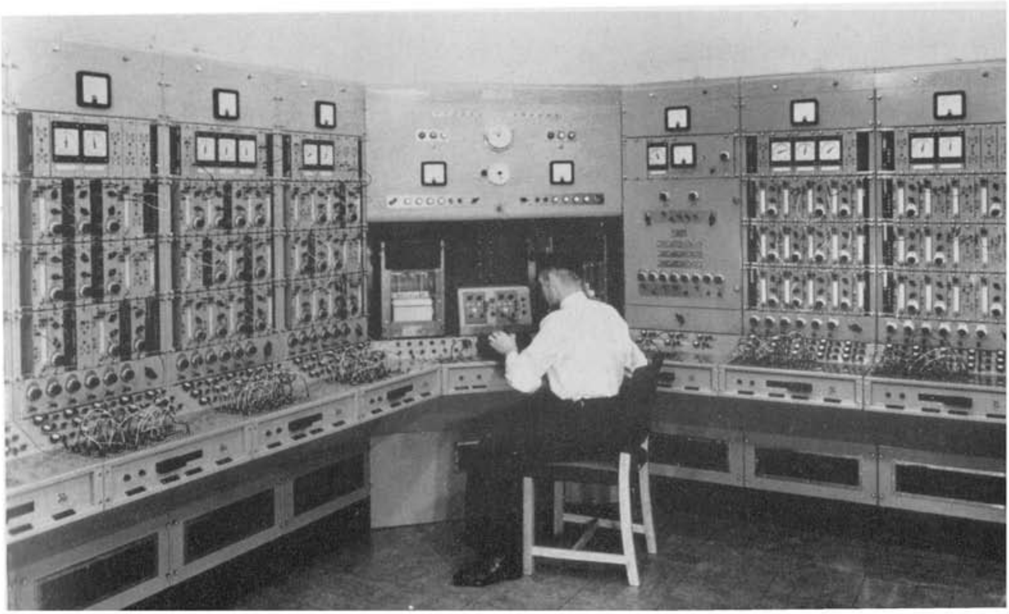


Plate 13 (a). Analogue computer system designed by E.M.I. Electronics as a simulator for studying complex control problems associated with guided weapons

Plate 13 (b). Nuclear power station performance is simulated by this analogue computing system installed by the G.E.C. Simon-Carves Atomic Energy Group



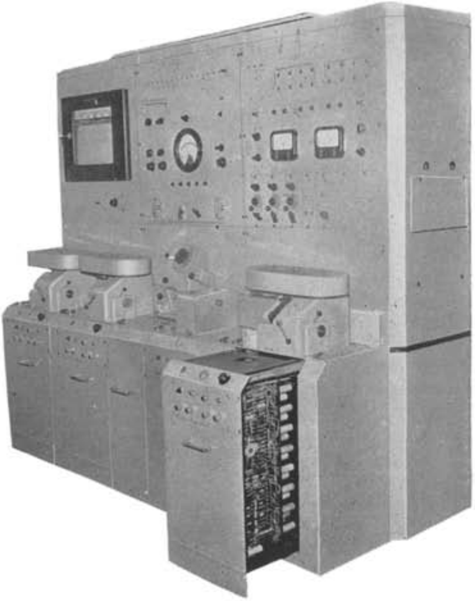
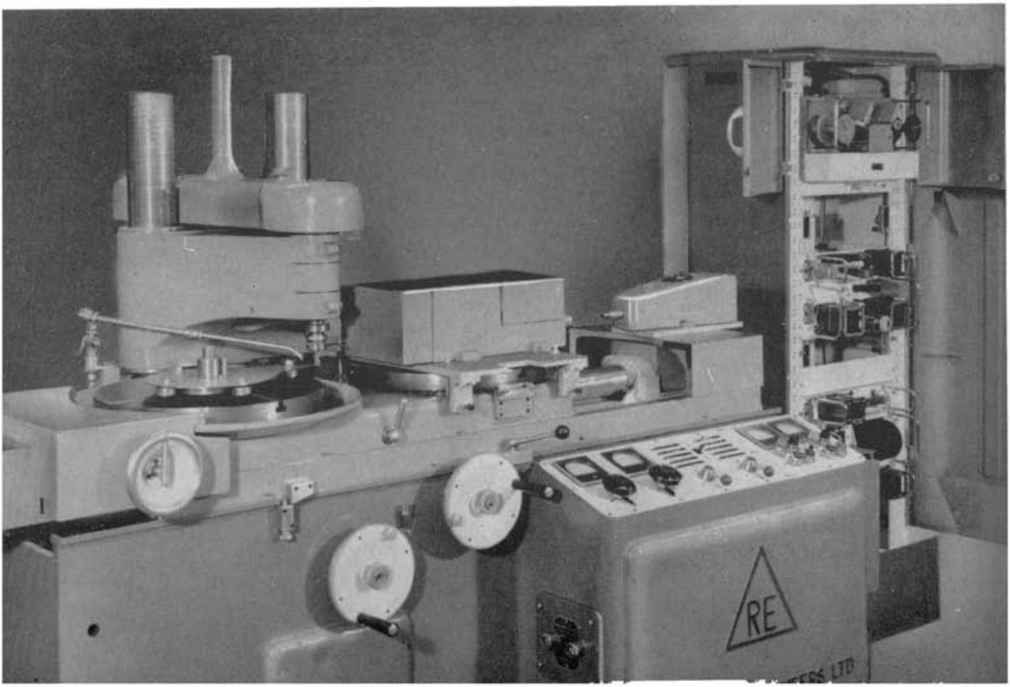


Plate 14 (a). Analogue computer for calculating correlation coefficients (Southern Instruments). The magazines for holding the continuous-line records can be seen about half-way up

Plate 14 (b). Machine for milling precision cams, with automatic control equipment (right) incorporating an analogue computer



than fifty different faults or changes in flight conditions. This panel injects the appropriate signals into the computers, which in turn transmit all relevant changes to the instruments in the flying control panels. The instructor can then assess the crew's reactions in appreciating the nature of the trouble and the appropriateness of the measures taken to rectify it.

Also behind the crew stations and to the left-hand side of the instructor are the radio navigational signal generators. Two flight recorders are provided which can be set up to correspond with four-course radio range signals or other radio navigational facilities, both at the point of departure and arrival of the synthetic flight. These recorders have circular charts, the centres of which represent the positions of the two beacons, and the track of the aircraft during flight is recorded by a moving pen. Altitude throughout the flight is registered on a separate drum recorder. Signal strength from the beacons is automatically varied according to the distance. Crosswinds and beacon failures can be presented to the crew with other flight problems from these panels.

Another type of flight simulator has recently been developed for training people to direct tactical air exercises from the ground by means of radar—the idea being that most of the aerial battles of the future will probably be conducted in this way. What the simulator does is to represent the various aircraft as responses on a synthetic radar screen, which is viewed by a pupil and his instructor. The signals representing the aircraft are produced by a group of control units which are manned by operators divided up between " enemy forces " and " friendly forces ". Each control unit is capable of simulating the performance of any type of aircraft (or group of aircraft) and is arranged so that this performance (in such things as rate-of-turn or rate-of-climb) cannot exceed the practical limits found in the real thing.

Associated with each control unit is a computer and this simulates the process of radar detection by translating the

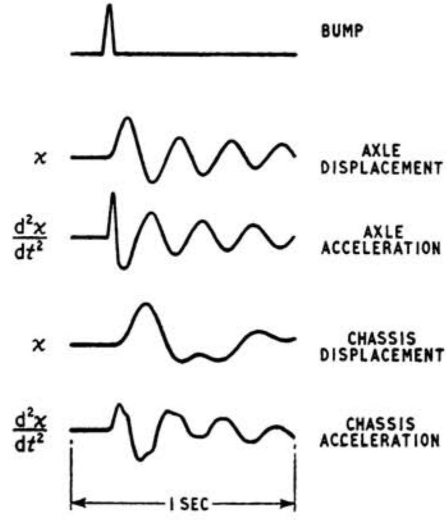


Fig. 6.2. Simulating the effects of uneven road surfaces on vehicle suspension systems. Graphs of voltages varying with time in an analogue computer used for this work

" aircraft's " course into range, bearing and height for presentation on the synthetic radar screen. All kinds of conditions that occur in practice are taken into account—the effects of winds on the aircraft, radar counter-measures such as " jamming ", variations in radio propagation and so on. It is therefore possible to stage a completely realistic air exercise, from the radar observer's point of view, without taking a single aircraft off the ground. Apart from the military applications, this type of simulator can also be used for training the aircraft controllers who act as " traffic policemen " at civil airports.

Descending from air transport to earth-bound road transport, the German automobile industry has recently been using an analogue computer as a research tool to simulate the effect of uneven road surfaces on vehicle suspension systems. Computing circuits are set up on the machine to represent the type of axle, suspension, shock absorber and chassis to be used and also the tyre pressure. Various types of bumps and holes in the road are represented by input voltage pulses, and the resulting displacements of the axle and of the chassis are given by output voltages varying with time. An example is shown in Fig. 6.2, where x represents displacement and d^2x/dt^2 acceleration (which is also required to find the stress in the parts). By adjusting the computer controls to alter the values of such parameters as tyre pressure, shock absorber damping, and stiffness of suspension springs, an arrangement

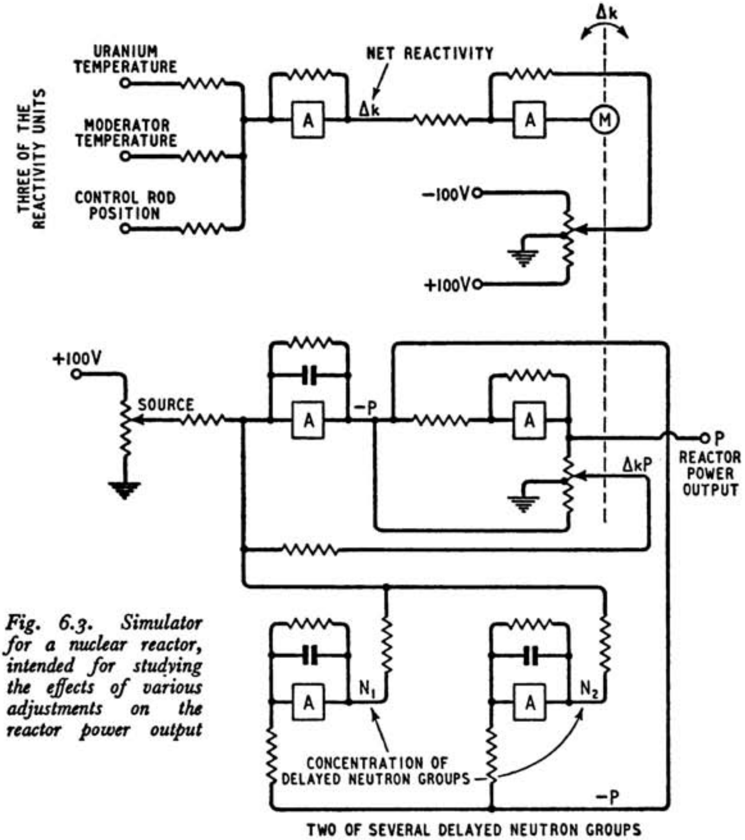


Fig. 6.3. Simulator for a nuclear reactor, intended for studying the effects of various adjustments on the reactor power output

can be found which gives the best overall performance. In this way, the best type of suspension systems for new vehicles have been determined in a fraction of the time normally required.

A very important recent application of analogue simulators has been in the design of nuclear power stations for the British nuclear power programme. Here it is necessary to simulate, in the first place, the nuclear reactor itself by setting up analogue circuits which govern its operation. A simplified example is shown in Fig. 6.3. It is then possible to see the effect of adjustments, such as movements of the control rods, on the reactor power output. The machine shown in Plate 13 (b) has one cabinet permanently connected to simulate the reactor itself while another six cabinets are general-purpose equipments which can be arranged to simulate the performance of heat exchangers, control rods, coolant circulators, turbo-alternators or other components of the complete power station. The main uses so far have been the study of transient effects produced by disturbances such as faults and the analysis of automatic control systems for the reactor.

It would not be fair to give the impression that the simulator is the only form in which the analogue computer is used for really practical purposes outside the laboratory. Straightforward analogue computing devices are beginning to appear in quite a number of industrial control and inspection systems. As an example, a machine has been developed by the British Rayon Research Association for analysing the irregularities which occur in the cross-section of yarn. The purpose of this is to discover any periodic components in the irregularities, for these cause highly objectionable patterning in a fabric when it is woven into cloth widths bearing some relationships to the wavelength of the periodicity.

To begin with, the variations of cross-section along a length of yarn are converted into electrical signals and recorded on a corresponding length of magnetic tape. This tape is " played back " to an analogue computer through two pick-up heads with a certain spacing between them, and what the analogue computer does is to measure the correlation between the cross-section values picked up by the two heads at a variety of different spacings. If the head spacing corresponds to the periodicity in the yarn, then the correlation will be high and the computer gives an indication of this. It actually plots a curve showing how the correlation varies with the pick-up head spacing, and any rhythmical fluctuation in this curve gives the periodicity in the yarn cross-section.

This type of analogue computer is known as an " autocorrelation computer ", because it finds the correlation between values occurring at different times in the same physical effect—or strictly speaking the extent by which a value at time t is proportional to a previous value at $t - \tau$. The measure of this proportionality is given by a mathematical expression known as the " autocorrelation function ", and this is what the machine actually computes. As a great many autocorrelation computers are used in various branches of science and technology, it is worth while seeing how this expression is derived and how the calculation is mechanized in the computer.

Suppose, then, that one has a graph showing the variations in time of some physical effect—for example, the output signal from one of the playback heads mentioned above giving the variations in yarn cross-section. If a certain value at time t on this graph is large it is more likely to be related to a previous large value at time $t - \tau$ than to be the result of some other cause. Moreover, if this previous value at $t - \tau$ is large it is more likely to be associated with a large value at t than if it is small. Thus the two values have a greater probability of being related when they are both large together than when one is large and the other is small. When they are both small it is not possible to draw any conclusions. One can say, then, that the correlation between any pair of such values is proportional to their product, which is only large and positive when both of them are large (either positive or negative) together. It is necessary, however, to take the average of a large number of such products to be able to decide the real extent of the correlation between the values.

In principle this averaging should be performed over all time, but in practice it is only necessary for the averaging time to be large compared with the slowest periodicity concerned. Thus, if one calls the two values $f(t)$ and $f(t - \tau)$ the autocorrelation function turns out to be the average of the product $f(t).f(t - \tau)$, and this is, in fact, the calculation which the analogue computer has to perform.

The first practical problem in dealing with an actual signal or effect varying in time is to devise a system whereby the earlier value occurring at $t - \tau$ can be presented to a multiplying circuit *at the same time* as the " present " value $f(t)$. A very popular method—and the one adopted for the yarn-testing device—is based on the use of magnetic tape recording. The variation in time of the physical effect is first of all recorded on the tape (see Fig. 6.4). One of the playback heads then gives $f(t)$ while the other, being displaced along the tape, simultaneously gives the value as it was recorded at an earlier time—that is, $f(t - \tau)$. A multiplying circuit then finds the product $f(t).f(t - \tau)$ and this is passed to an integrating circuit which computes the average over a period of time—that is,

$$\int f(t).f(t - \tau)dt.$$

For investigating periodicities it is necessary to be able to vary the time lag from zero upwards, and in the computer this is done in steps by an automatic mechanism which alters the distance between the playback heads for each successive revolution of the tape. Then, for each particular value of τ

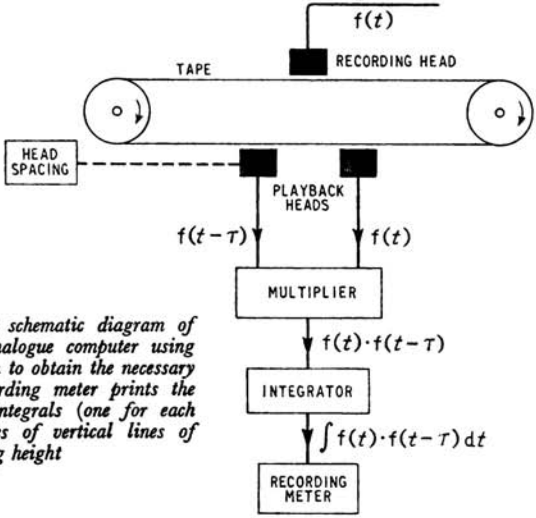


Fig. 6.4. Simplified schematic diagram of an auto-correlation analogue computer using a magnetic tape system to obtain the necessary time lags. The recording meter prints the values of successive integrals (one for each time lag) as a series of vertical lines of varying height

the value of the averaged product is plotted by a recording pen, usually as a vertical deflection from an horizontal zero line. The envelope enclosing all the successive deflections forms a curve which shows how the correlation between the two values is related to their time separation, and from this any periodicities in the physical effect can be read off.

Analogue computers of this kind have been used for analysing the periodicities in a great many different physical phenomena—speech waveforms, meteorological data and electrical brain rhythms, to name a few. Without such mechanical aid the necessary calculations would be extremely laborious and, in fact, would probably never be done at all. Plate 14 (a), p. 86, shows a commercial correlation analogue computer, made by Southern Instruments, which has photo-electric line followers to work from continuous-line records on film or paper.

In Chapter 1 it was mentioned that one of the main applications of computers in the field of " automation " at present is the automatic control of machine tools. Here the general principle is that the cutting movements of, say, a milling machine are controlled by information fed in by a magnetic or punched paper tape. This information consists of a series of co-ordinates of separate points defining the contour to be cut, and it is the job of the computer to interpolate between these points. In this way the amount of information which the human " planner " has to put on the tape is restricted to a relatively small number of " marker " points.

A system of control using an analogue computer for the interpolation has been devised by E.M.I. Electronics and applied to several types of machine tools. One, for cutting precision cams, is shown in Plate 14 (b), p. 86. The input information is supplied by a punched paper tape and the " marker " points obtained from it are actually successive radial dimensions defining the outline of the cam.

The tape is fed into the machine in synchronism with the rotation of the work-table so that for each new " marker " radius the work is brought to the appropriate angular position. Several successive values from the tape are held temporarily in a storage system and thereby are made available simultaneously (in the form of voltages) so that they can be applied to the computer for interpolation. The computer then produces voltages representing radial dimensions of a parabolic curve between the " marker " points, and these are used to control a mechanism which moves the work-table longitudinally relative to the cutting tool.

This use of an analogue computer as part of an electronic control mechanism is, in fact, typical of a good many " automation " techniques. The computing apparatus is very specialized in function and forms an integral part of the complete equipment, so that it can hardly be recognized *as* a computer.

Another machine falling into this category is a computer designed by Sargrove Electronics for the control of continuous manufacturing processes. It works on statistical principles and makes corrections to the processing machinery on the basis of error trends which it detects in the finished product.

The apparatus (Fig. 6.5) includes a mechanical measuring device to sense the variations in the product, a transducer to turn these into electrical signals, a selecting device which samples the product at suitable intervals and filters out unwanted information, and a " deviation classifier " which quantizes the error information into three distinct categories —" positive error ", " no error " and " negative error ". A statistical analyser then computes the sum of the errors, retaining in storage their net sense, and actuates correction timers (positive or negative, as appropriate) which by their time of operation control the amount of correction applied to the processing machinery.

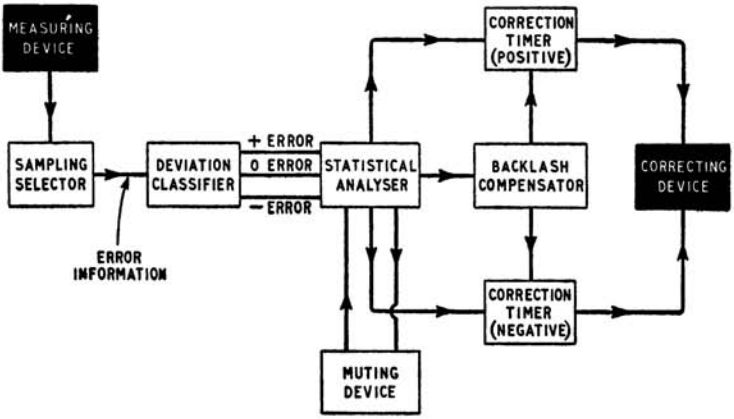


Fig. 6.5. Automatic control system using a computer working on statistical principles

A " muting circuit ", consisting of a timer, disconnects or paralyses the statistical analyser until the corrected product is itself being sampled, while a " backlash compensator " increases the operating time of the timers if there is a change of sense in the correction.

One particular application of this statistical computer has been in an apparatus for controlling the weight of pieces of dough in the mass-production of bread. This is very necessary, of course, to ensure that the final loaves are not underweight. Each piece of dough is weighed in a high-speed weighing machine and its error is quantized into " light ", " heavy ", " very light " or " very heavy ". The computer then calculates the error trend and makes suitable corrections by controlling the size of the cavity in which the dough is moulded.

7

DIGITAL COMPUTER CIRCUITS—1

BASIC ARITHMETIC OPERATIONS

THERE is no denying that an electronic digital computer is an impressively complex piece of apparatus when viewed as a whole. Nevertheless the individual circuit " bricks " from which it is composed are quite simple—indeed simpler than those of an average radio receiver or television set. The apparent complexity comes from the fact that a complete machine contains a large number of such " bricks " and the connections between them often form a very elaborate and intricate pattern. In this way the digital computer bears a superficial resemblance to the animal nervous system. The nerve cells are individually quite simple as functional units but the versatility of the system as a whole results from the complexity of interconnections.

The reason for the basic simplicity of the digital computer circuits is, of course, that numbers are represented in the binary scale, as already explained in Chapter 2. Only two electrical states are required, " off " and " on ", as against the ten states needed for the decimal scale. Thus the binary number 1101 might be registered in static form by a row of four valve or transistor circuits acting as electronic switches as shown in Fig. 7.1 (a). Working from the left, the first two would be " on ", the third would be " off " and the fourth " on ". Alternatively, the number might be represented in dynamic form by a negative voltage, say −50 V, being either switched on (for " 1 ") or switched off (for " 0 ") at particular instants of time. Applied to a single wire, this would give a train of negative pulses as shown in Fig. 7.1 (b).

It will be noticed here that the order of digits of the 1101 number has apparently become reversed to 1011. This is simply because of the convention in graphs that time increases from left to right. In fact, of course, the number is not reversed on the time scale and is represented quite logically, with the least significant digit occurring first, then the next significant, and so on.

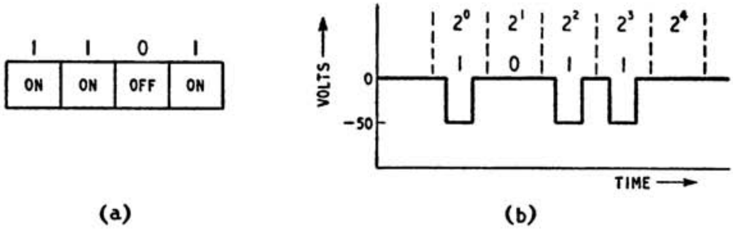


Fig. 7.1. Representing a binary number (a) in static form by a row of electronic switches, and (b) in dynamic form by a train of pulses

Note that the respective time intervals from the left, in which pulses either appear or do not appear, correspond to numerical values of 2^0, 2^1, 2^2, 2^3, etc. (that is, 1, 2, 4, 8, etc.), the complete decimal number represented being 13.

Having seen how the binary numbers are represented by electrical states or pulse signals, we can now pass on to consider how these can be manipulated by circuits to perform arithmetic operations. To begin with, take the simplest case of addition. We have already gathered from Chapter 2 that the two significant rules of binary addition are: $0 + 1 = 1$ and $1 + 1 = 0$ and carry 1. In order to see clearly what we want to do it is a good idea to write down the rules in more formal style as follows:

Condition A $0 + 0 = 0$
Condition B $1 + 0 = 1$
Condition C $0 + 1 = 1$
Condition D $1 + 1 = 1 \leftarrow 0$ (or 10)

All that is necessary, then, is to devise a circuit that will give outputs corresponding to the right-hand column of answers when inputs corresponding to the two left-hand columns of numbers to be added are applied to it. We may not be able to follow the arithmetic operations of such a circuit in the same way as we can, for example, the operations of a decimal adding machine, but that is simply because our minds are more accustomed to decimal notation. As long as the circuit obeys the binary rules, however, we can be sure that it will produce the right answers, and that is all that really matters.

Supposing we are working with pulses to represent the binary digits, there are two basic elements required to build up the adding circuit. These are called "gates", as they perform switching operations equivalent to opening or closing gates to applied pulses. The first, which is shown in Fig. 7.2, is known as an "OR" gate, as it has the property of giving a pulse at the output when a pulse is applied to one input *or* the other (or, incidentally, when pulses are applied to both). It consists of a high-value resistor R and two diodes, which can be either thermionic or semiconductor types. The circuit is arranged so that in the quiescent state a current is flowing from the positive supply through R and dividing equally through the diodes in their forward direction. When a negative pulse is applied to one of the inputs the output point also goes negative, because the resistance of the diode is low compared with R. The other diode, however, has its upper electrode driven negative by the output voltage and so ceases to pass current;

consequently the output is entirely determined by the pulse applied to the first diode. Additional inputs can be connected as shown by the broken line. A functional symbol for this gate is given at the top of the diagram.

The other basic element is known as an " AND " gate, because it gives a pulse at the output only when a pulse is applied to one input *and* to the other simultaneously (or to all inputs simultaneously when there are more of these than two).

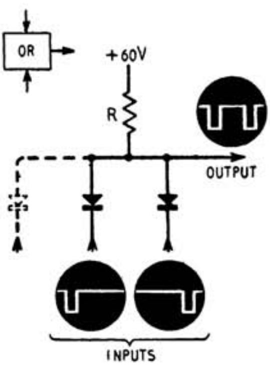


Fig. 7.2. Gate circuit used for the " OR " logical operation

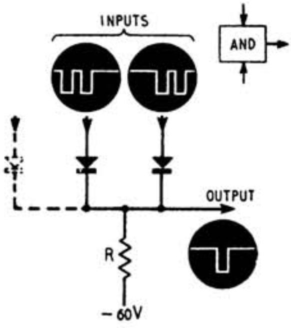


Fig. 7.3. Gate circuit employed for performing the " AND " logical operation

Here the circuit is shown in Fig. 7.3. As before, in the quiescent state a current is flowing through R and dividing equally through the diodes. Only when negative pulses are applied to the diodes simultaneously and they *both* cease to conduct, is the total current through R reduced so that a negative pulse is produced at the output point. If a negative pulse is applied to only one of the diodes, the current which was flowing through it now passes through the other one, so that the total current remains unchanged from the quiescent state and no pulse is formed at the output. Again the manner of connecting additional inputs is shown by the broken line, while a functional symbol for the gate appears at the top.

One can now forget about the actual circuit operation of these gates and just consider them as functional units. How, then, can they be arranged to put into effect the rules of addition set out on p. 99? Looking at Conditions B and C, it is fairly obvious that a simple " OR " gate will satisfy them—and incidentally it will also satisfy Condition A because if no pulse is applied to either or both of the inputs no pulse will be given out. This is illustrated at the top of Fig. 7.4, with x and y for the inputs and the sum coming out of the " OR " gate. The real difficulty is with Condition D. Here it is necessary to get no pulse at the output when a pulse is applied to both inputs. And at the same time a pulse must be generated on another wire to represent the 1 carried into the next column.

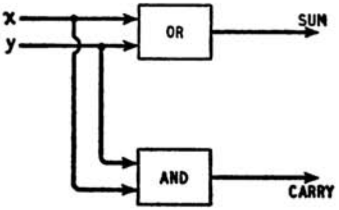


Fig. 7.4. Basis of a circuit for adding two numbers, x *and* y

Taking the last thing first, the carry pulse can be produced by connecting an " AND " gate to the input wires, in parallel with the " OR " gate, as shown in Fig. 7.4. Thus if 1 + 1 is applied to x and y, a 1 appears on the output " carry " wire, and that part of Condition D is satisfied. But as a result of the two pulses on x and y there is also a pulse fed to the " sum " output (because the " OR " gate has allowed it through) and this is definitely not wanted. It only remains, then, to devise some means of cancelling this 1 to give a 0 on the "sum" output whenever a 1 appears on the "carry" wire.

This can be done by a special kind of "AND" gate which is normally open, so that a negative pulse on one of the inputs is allowed to pass through to the output, but is closed whenever a negative pulse is applied to the other input. The circuit consists of an ordinary "AND" gate with a valve or transistor inserted in the second input to act as an inverter to the negative pulses. A negative pulse appearing on the second input is thereby turned into a positive-going pulse, and is used to cancel any negative pulse on the first input with which it happens to coincide.

Fig. 7.5 shows the arrangement. Here the diode D_1 corresponds to one of the diodes in the normal "AND" circuit, Fig. 7.3, while the triode valve and its resistor network take the place of the other diode. Resistors R_1, R_2 and R_3 are arranged so that when no negative pulse

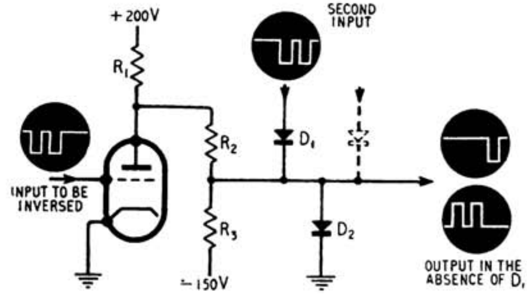


Fig. 7.5. Special type of "AND" circuit with an inverter in one of the inputs

is applied to the grid of the triode, the junction of R_2 and R_3 is at a negative potential. Diode D_1 will therefore normally pass current until a negative input pulse is applied to its upper electrode, when it will cease to conduct and so produce a corresponding negative pulse at the output. If, however, a negative pulse is applied to the grid of the triode, the inverting effect of the valve will drive the R_2R_3 junction point—and the lower electrode of D_1—in a positive direction. This will apply a reverse voltage to D_1 and will inhibit its normal action, in that any additional reverse voltage applied from the other side, as a negative input pulse, will have no further effect. Thus, in this condition there will be no pulse at the output. The purpose of D_2, incidentally, is to prevent the voltage at the R_2R_3 junction from rising above earth potential (corresponding to a " 0 ") when it goes in a positive direction.

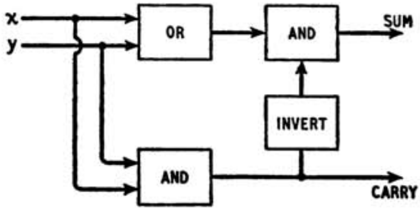


Fig. 7.6. Arrangement of a " half-adder " or " two-input adder " derived from Fig. 7.4

With this modified " AND " gate it is now possible to complete the adding circuit, and the final result is shown in Fig. 7.6. This obeys all the rules set out on p. 99, but is lacking in one respect: it does not have any means of accepting a " carry " at the input which may result from the addition of the less significant digits of two numbers. For example, when one goes through the motions of adding 39 + 26 like this

$$\begin{array}{r} 39 \\ +\ 26 \\ \hline 65 \end{array}$$

one says " six plus nine equals fifteen; five down and carry one ". Then " two plus three, *plus one carried*, is six ". Not only does one have to add 2 + 3 to form a temporary answer of 5, but also take the 1 carried over and perform a second addition with that. The circuit of Fig. 7.6, then, has no means of taking a " carry " digit at the same time as the *x* and *y* inputs. For this reason it is known as a " half-adder " or sometimes as a " two-input adder " and the functional diagram is shown in Fig. 7.7 (a).

The limitation is quite easily overcome, however, simply by arranging an identical two-input circuit to add the " carry " digit to the output of the first half-adder. This is shown in Fig. 7.7 (b), the " carry " input being represented by *z*. The " OR " gate is required to take care of any " carry " at the output which may result from adding in the extra digit at *z* when *x* and *y* by themselves do not produce one. Thus we have arrived at a complete adding circuit, sometimes known as a " three input adder ", and this can be drawn in the simplified form shown at (c). Incidentally, the same results can be achieved by other arrangements of gate circuits.

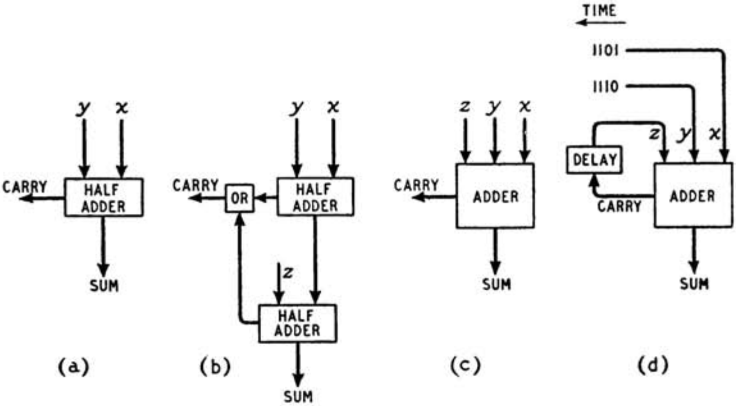


Fig. 7.7. Functional diagrams of (a) half adder, (b) complete or three-input adder with simplified form at (c), and (d) an adder with a storage arrangement for holding the " carry " output

It only remains now to see how the adder will deal with two binary numbers presented to the *x* and *y* inputs as pulse trains of the kind shown in Fig. 7.1 (b). Supposing that 1101 is to be added to 1110 (13 to 14 in decimal), the operation would be set out on paper as follows:

$$\begin{array}{r} 1101 \\ +\ 1110 \\ \hline 11011 \end{array}$$

In the adding process each pair of digits is taken in turn, working from right to left—both on paper and with the corresponding pulse trains presented to the actual circuit. The least significant pair of digits is quite straightforward and also the next pair. With the third pair, however, a " carry " is formed. On paper this carry digit merely has to be taken into the next higher column, but in the actual circuit the corresponding pulse has to be delayed in time until the next pair of digit pulses arrives at the x and y inputs.

This is done as shown in Fig. 7.7 (d), by passing the " carry " output through a delay device with a delay time equal to one digit period (i.e., the interval between successive pulses). Thus the pulse is held back until it can be added, via input z, to the sum of the next higher value pair of digits arriving at x and y. Another " carry " is formed in the process and again the resulting pulse is delayed by one digit period. But this time when it arrives at z there is nothing else for it to be added to, and so it simply appears at the " sum " output to give the final " 1 " in the fifth position.

In most digital computers the two numbers to be added do not become available simultaneously as a matter of course. Usually one of them has to be held ready until the other arrives. This is done by the process of *storage*, and various techniques by which it can be achieved are described in Chapters 8 and 9. It has already been mentioned at the beginning of this chapter that a four-digit binary number like 1101 can be represented in static form by a row of four two-state valve circuits. In the same way, such a row of circuits can be used to store a number which is presented to it as a train of pulses, as in Fig. 7.1 (b)—the first " 1 " pulse being arranged to switch " on " the extreme right-hand circuit, the following " 0 " space to leave " off " the next circuit and so on, working leftwards. In effect the pulse train is made to " stand still " in the manner already shown in Fig. 7.1 (a). A row of two-state circuits, or any other storage device which holds a complete number in the fashion, is called a *register*—a term derived from mechanical calculators.

Having stored one of the numbers in such a register, the next problem is how to feed it out again into the adder when the other number arrives. For this purpose a special

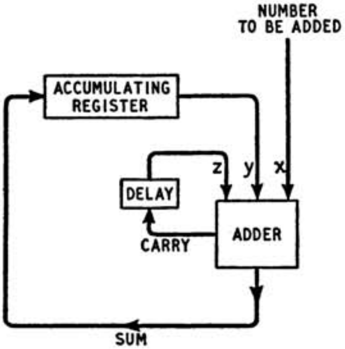


Fig. 7.8. Adding system in which a running total is accumulated in a register

type of register is used which circulates its contents through the adder, as shown in Fig. 7.8. The stored number passes digit by digit to the y input of the adder and, if nothing is applied to the x input, it emerges from the " sum " output unchanged and returns to the other end of the register. Thus the complete pattern of digit pulses moves continuously round the loop " chasing its own tail " so to speak. For example, if the number concerned is 1101, the pattern in the register goes through the following cycle: 1101, 1110, 0111, 1011, 1101 . . . and so on.

When the other number is applied to the *x* input, it is added, digit by digit, to the circulating number from the register. A new pattern of digits then emerges from the " sum " output and takes the place of the original single-number pattern in the register. If yet another number were applied to *x* it would be added to the first total, and so on. Thus the register is arranged so that it accumulates a running total, adding new numbers to whatever is already in it. For this reason it is called an *accumulating register* or, more commonly, just an *accumulator*. In some computers the number in the accumulator simply remains stationary until it is required to move. In others it continually circulates through the adder. However, the actual circuit details of the device, and the way in which the two number pulse-trains are synchronized so that corresponding pairs of digits are presented together, will be dealt with later.

The accumulator can also be used in conjunction with other arithmetic circuits for doing subtraction, multiplication and division, always storing the result of each successive operation. It will be fairly obvious that a subtracting circuit, for example, can be constructed from electronic gates in much the same way as the adding circuit. First of all the rules for binary subtraction are written down as a table of conditions, as on p. 99, then appropriate gates are connected together to satisfy these conditions. Sometimes, however, the adding circuit alone is used for the other arithmetic operations. Subtraction can be achieved by adding complements of numbers, multiplication by successive addition, and division by the reverse process of repeated subtraction.

The subtraction method using complements of numbers is much the same in binary notation as in decimal—and perhaps simpler in some respects. For example, suppose that we wish to do the decimal subtraction

5823 − 2174. A well-known trick is to take the "nines complement" of the subtrahend 2174, then add 1 to give the true complement and finally add this to the 5823, ignoring any fifth digit which may come out in the answer. Thus the nines complement of 2174 is 7825 (each digit being subtracted from 9) and 1 added to this makes 7826. Then 5823 + 7826 = (1)3649 which gives the answer 3649 correctly.

Of course, this is needlessly complicated for ordinary decimal arithmetic but it serves to illustrate the binary method. Suppose, then, that we wish to do the binary subtraction 10010 − 01100 (or 18 − 12 in decimal). The complement of the subtrahend 01100 is obtained simply by reversing all the digits (since there are only two in binary) to give 10011. In this case the 1 is not added immediately but first of all the operation 10010 + 10011 is performed to give (1)00101, then it is the unwanted (1) on the left-hand side which is transferred to the right and added. Thus we get the correct answer 110 (or 6 in decimal). In practice it is quite a simple matter to obtain the complement by connecting an inverter circuit (see Fig. 7.5) in one input of the adder, thereby changing all pulses into spaces and spaces into pulses. Transferring the unwanted 1 from left to right and adding it is equally straightforward.

The process of multiplication can, as already mentioned, be performed by repeated addition of the multiplicand. Thus the product of 101 × 100 (or 5 × 4) can be obtained by adding the 101 four times to the contents of the accumulator (initially zero) like this:

$$000 + 101 + 101 + 101 + 101 = 10100 \text{ (decimal 20)}$$

For this purpose, of course, a counter is required to control the number of addition cycles. A more common method of multiplication is based on the fact that moving the digits of a binary number one place to the left multiplies that number by 2. For example 1011 (or 13) moved one place to the left becomes 10110 (or 26). This happens, of course, because binary numbers are formed on the base 2 and the respective digit positions stand for 2^0, 2^1, 2^2, 2^3, 2^4 and so on. The process is equivalent to moving a decimal number one place to the left (and adding 0) when we multiply by 10. For example, if we wish to multiply a number by 432, we first of all perform × 2, then move the number one place to the left and perform × 3 to represent multiplying by 30, move it another place to the left and perform × 4 to represent multiplying by 400, and finally add up the partial products.

Similarly, if we wish to multiply by a binary number—say 1011, taking the last significant digit first, this means multiplying by 1×2^0, then 1×2^1, then 0×2^2 and finally 1×2^3. Thus, if the multiplicand is, say, 1100, it works out like this:

$$\begin{array}{rl}
1100 & \text{(multiplicand)} \\
1011 & \text{(multiplier)} \\
\hline
1100 & (1 \times 2^0) \\
1100 & (1 \times 2^1) \\
0000 & (0 \times 2^2) \\
1100 & (1 \times 2^3) \\
\hline
10000100 &
\end{array}$$

This example shows that the operation consists of nothing more than successively moving the multiplicand to the left, adding it where the multiplier digit is a " 1 " and *not* adding it where the multiplier digit is a " 0 ". In the actual computer, where the numbers are represented by pulse trains, the arithmetic process of " moving one place to the left " means delaying in time by one digit-pulse period. A glance back at Fig. 7.1 (b) will make this clear.

Thus the essential parts of a multiplying circuit based on this principle are a delay device and an adder with an accumulator, as in Fig. 7.8, to allow the shifted multiplicand to be added each time to a running total. A type of gate circuit is used to test each digit of the multiplier in turn; if it is a " 1 " the shifted multiplicand is added and if it is a " 0 " the multiplicand is not added.

It can be seen that this process of multiplication is somewhat slower than, say, simple addition, and in computers where a great deal of it has to be done it is more usual to build special high-speed multiplier circuits. These are faster in operation partly because they use more than one adder (actually one for each digit) and partly because the numbers are presented in " broadside ", with all digit pulses appearing simultaneously instead of in serial form. A similar situation holds with the operation of division, and special dividing circuits are quite often built into computers.

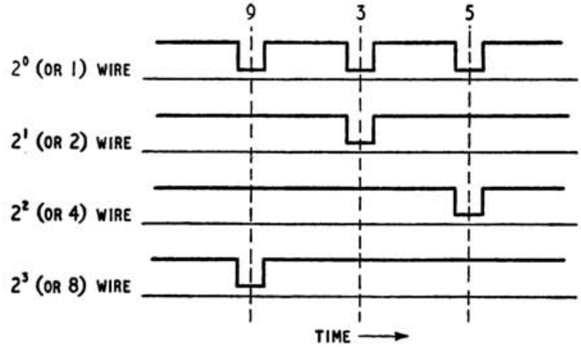


Fig. 7.9. Representation of the number 539 in binary pulse form on four wires by the 1,2,4,8 code

Before leaving the subject of arithmetic operations, it should be mentioned that some of the smaller digital computers do not operate in pure binary notation but in binary-coded decimal. Here each digit of the decimal number is represented by the equivalent binary form, as already explained in Chapter 2, the main idea being to avoid the extra complications necessary for decimal/binary conversion inside the machine. With this system it is quite usual for each decimal digit to be presented as a pattern of four simultaneous pulses or spaces on four parallel wires. Thus the number 539 would appear as in Fig. 7.9, the "9" pattern arriving first, then the "3" pattern and finally the "5" pattern.

This method of representing numbers is sometimes called the "1, 2, 4, 8 code"—from the fact that the four wires carry pulses of value 2^0, 2^1, 2^2 and 2^3 respectively. In order to perform, say, addition on such numbers it is necessary to have a separate adder for each of the four wires—one for the "1s", one for the "2s", one for the "4s" and one for the "8s". The essentials of the complete adding circuit are shown in Fig. 7.10. There are four output wires, giving a binary-coded decimal digit equal to the sum of the two coded decimal digits applied to inputs *x* and *y*. If a carry is formed by the "1s" adder it is transferred to the "2s"

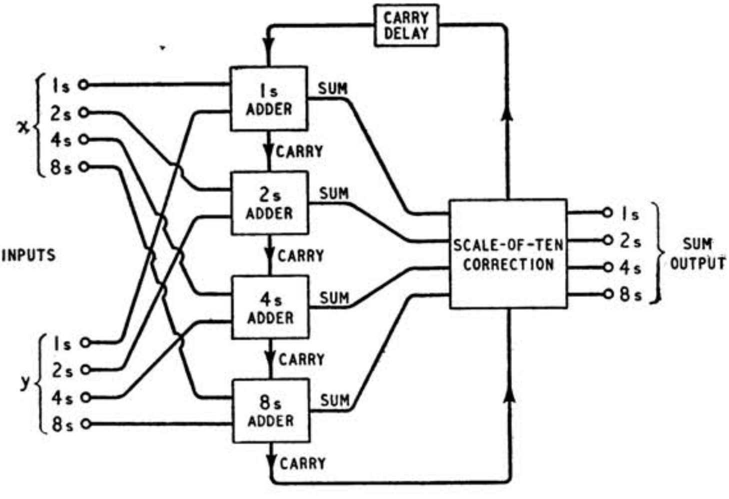


Fig. 7.10. Adding circuit for binary-coded decimal numbers of the kind illustrated in Fig. 7.9

adder, if a carry is formed there it goes on to the "4s" adder, and so on.

The system, however, only works straightforwardly when numbers up to 9 are produced at the sum output. For example, if the number 539 in Fig. 7.9 is being added to, say, 124, the first operation is to add the binary-coded versions of 9 and 4, which arrive simultaneously at the x and y inputs. This is 1001 + 100, which comes to 1101 (decimal 13), producing a 1 from the "1s" adder, a 0 from the "2s" adder, a 1 from the "4s" adder and a 1 from the "8s" adder. But in decimal arithmetic we do not want this complete 13 at the output. The correct procedure is to put down 3 and carry 1 into the "tens" column. Clearly, then, some kind of correction has to be applied to the Fig. 7.10 adder to make it work to the rules of decimal arithmetic.

This is where the "scale-of-ten correction" block enters into the picture. What it does is to add the binary-coded version of 6 to the sum whenever this sum exceeds 9. Thus, in the example above the sum 1101 (decimal 13) has 110 (decimal 6) added to it to produce 10011, which is a five-digit binary number. As a result the left-hand 1 becomes a carry. But the carry cannot be added to the next pair of binary-coded decimal digits until they actually arrive at x and y in the next pulse period, so it has to be delayed till then in the "carry delay" block. This next pair of digits (3 and 2 in the example) represent tens in decimal notation, and the carried 1 which is added to them represents the 1 ten obtained from the earlier summation of 9 and 4. Having disposed of the carry successfully, the complete adder has 0011 left on its output wires, and this of course represents the required 3 which has to be "put down" in decimal arithmetic.

A carry may or may not be produced by the scale-of-ten correction, but there will be a carry in any case from the "8s" adder if the sum of the binary-coded decimal digits applied to x and y exceeds the maximum four-digit binary number 1111 (decimal 15). Thus if 1000 and 1001 (decimal 8 and 9) are applied to x and y the addition gives 10001 (decimal 17), a five-digit binary number. The resulting carry then has to be delayed until the next pair of digits arrives at x and y. This leaves 0001 on the sum wires, which, as already explained, has to have 110 (decimal 6) added to it as scale-of-ten correction to give the required 111 (decimal 7 " put down ").

The parallel or " broadside " method of operation described above is also used to some extent in digital computers working in pure binary notation. It has the advantage of giving much faster arithmetic operations than the serial method of taking each digit of a number in turn, but requires a good deal more equipment, as can be seen from Fig. 7.10. However, if the digital computer is required for use as, say, part of a control system operating in " real time ", this extra equipment will probably be essential to achieve the necessary speed of operation.

8

DIGITAL COMPUTER CIRCUITS—2

CONTROL AND GENERAL ORGANIZATION

THE last chapter dealt with arithmetic circuits merely as isolated pieces of apparatus, performing isolated operations. In a practical computation, however, we usually want to do a whole series of these basic operations. For example, when faced with a calculation something like this:

$$54{\cdot}8\,(739-251) + \frac{257}{39{\cdot}6}$$

we can only deal with it by breaking it down into a sequence of simple operations. In this case this would run as follows: (1) perform $739 - 251$, (2) store the sum, (3) multiply this sum by 54·8, (4) store the product, (5) perform $257 \div 39{\cdot}6$, (6) store the quotient, (7) add this quotient to the product stored at stage (4).

In the actual computer we want this sequence of operations to be carried out automatically, with all the numbers produced at just the right times to be operated upon. The great speed of the electronic arithmetic circuits would obviously be completely wasted if a human operator had to connect them up, feed in the appropriate numbers and write down the partial results on paper at every single stage of the calculation. To achieve the desired automatic operation, therefore, the arithmetic circuits have to be suitably co-ordinated and synchronized—and this is the function of what is generally called the *control* section of the computer.

The reader will have already gathered from Chapter 2 that the control circuits themselves are actuated by a " programme " of instructions, prepared in advance by a human operator. These instructions are similar in form to the sequence of operations (1) to (7) set out above, and are fed into the computer and stored there as a series of electrical code patterns. In most digital computers the numbers to be operated on, or *operands*, are also stored in readiness for use (although in small machines it is not unusual for them to be fed in manually as required at each stage of the calculation).

The general mode of operation of the control circuits, then, is to scan the coded instructions in turn and use each one to set up a pattern of electrical connections which causes a particular operation to be carried out. In the smaller digital computers the coded instructions are stored directly in the form of patterns of connections, set up manually by inserting plugs into plug boards (as used in analogue computers). Thus no interpretation is necessary, and the computer is arranged simply to attach itself in turn to each set of connections. With larger machines, however, the coded instructions are usually in the form of stored pulse trains and have to be translated into patterns of connections. This is done by means of electronic switches or gate circuits of the kind described in Chapter 7, which, of course, can be opened or closed simply by the application of suitable voltages.

In this chapter we are particularly concerned with this larger " stored programme " type of computer—how it selects the coded instructions in the right order and converts them into switching arrangements. We are also mainly concerned with the type of computer in which numbers and instructions are represented by serial pulse trains—not the " parallel " kind mentioned at the end of Chapter 7. To begin with, however, it will be necessary to look at some of the actual circuit techniques by which the instructions and numbers are stored inside the machine.

A very common form of storage (and actually the first to be used in electronic computers) is the row of two-state electronic circuits, or static register, already mentioned in Chapter 7. The two-state circuit most frequently used for this kind of store is the Eccles–Jordan trigger circuit (see above), using either valves or transistors. Two of these circuits based on valves, from a ten-digit storage register, are shown in Fig. 8.1. Taking the one marked T_0 as

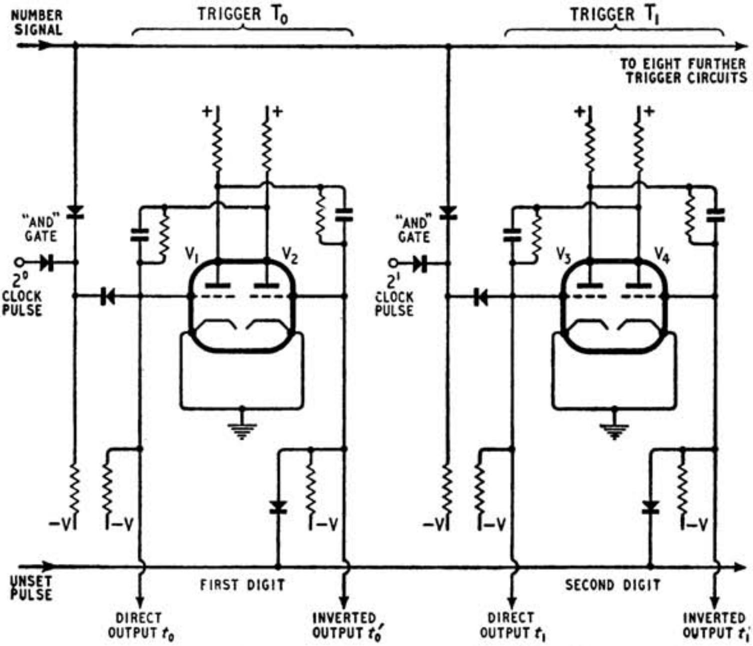


Fig. 8.1. Two Eccles-Jordan trigger circuits, T_0 and T_1, being part of a trigger-circuit store. By means of the " AND " gates and clock pulses the pulses of the incoming number signal are distributed to the appropriate triggers. When a " 1 " is stored the " direct output " of a trigger is negative and the " indirect output " is at about earth potential. When a " 0 " is stored the reverse applies

an example and ignoring the diode gates for the moment, the two states (which are both stable) are actually represented by: V_1 conducting with V_2 cut-off; V_2 conducting with V_1 cut-off. Transition from one state to the other is obtained by applying a negative pulse to one of the grids.

For example, suppose to begin with that V_1 is conducting and V_2 is cut off. If a negative pulse is applied to the grid of V_1 it will tend to reduce the anode current and so cause the anode potential to rise. This will drive the grid of V_2 positive and more current will flow in the valve, causing the anode potential to fall. As a result of this fall a negative voltage will be applied to the grid of V_1 which will augment the negative voltage already there from the pulse. A cumulative action will then take place round the loop, causing the circuit to be switched over rapidly to the other stable state with V_1 cut-off and V_2 conducting. If now a negative pulse is applied to the grid of V_2 the same action will take place in reverse and the circuit will therefore be switched back to its original stable state.

When V_1 is put into the cut-off condition by application of a negative pulse to its grid, the trigger circuit is said by convention to be " set " and registering a " 1 " digit. Conversely, when V_1 is conducting the circuit is said to be " unset " and registering a " 0 " digit.

It will be noticed from Fig. 8.1 that two outputs can be obtained from the two anodes of each trigger circuit, a " direct output " and an " inverted output ". When a " 1 " is stored the " direct output " is a negative voltage and the " inverted output " is at about earth potential. When a " 0 " is stored the reverse applies.

Transistors can be connected to form trigger circuits in much the same way as the valves in Fig. 8.1. An example of such a transistor two-state circuit is given later in Chapter 13.

Another commonly-used storage device is of the kind already mentioned in Chapter 1 in which a train of digit pulses is kept circulating in a loop consisting of an acoustic delay-line with its output connected back to its input. This, of course, is dynamic storage, as distinct from the static storage of Fig. 8.1. The delay-line itself often takes the form of a length of nickel wire. Pulses of current representing the digits are passed through a coil surrounding one end of the wire. These set up pulses of mechanical stress in the nickel by the magnetostriction effect, which travel at the speed of sound (1,100 ft/sec) along the wire. A receiving coil is placed at the other end of the line and has a permanent magnet near it. The arriving stress pulses produce changes of permeability in the nickel wire and these in turn produce changes in the magnetic flux linking the receiving coil. The resulting e.m.fs induced in the coil are then amplified and reshaped to give suitable signals for feeding back to the input.

Fig. 8.2 shows the general arrangement of a delay-line register. The line itself is made to have sufficient delay

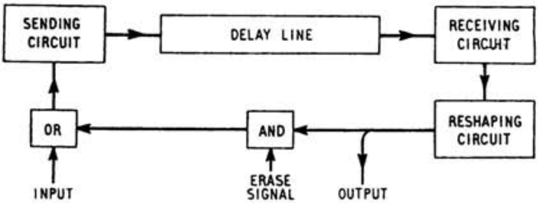


Fig. 8.2. Delay-line storage register. The " AND " gate is a modified type (Fig. 7.5) which closes when an " erase " signal is applied to it

to accommodate a train of digit pulses representing a complete number—in other words, the first pulse of the train does not emerge at the receiving circuit until after the last pulse has been passed into the line by the sending circuit. An " AND " gate with an inverter (Fig. 7.5 in the previous chapter) prevents the output of the delay-line from going back to the input when an " erase " signal is applied. If a new train of digit pulses is then applied to the following

" OR " circuit, it will replace the original one in the line. Sometimes an adding circuit is used instead of the " OR " gate. This does not affect the normal operation of the register, but if a new number is applied to the input it is added to the number already in the line and the sum is stored. In fact, the device becomes an accumulator, working on the principle described in the previous chapter.

When a binary number is circulating in such a delay-line store it is " chasing its own tail " and so becomes just a continuously repetitive pattern of pulses and spaces. Consequently there is no means of determining its actual numerical value unless one knows which pulse or space in the pattern is the first one in the train—representing the least significant digit. Obviously some means of identifying the true positions of the pulses and spaces is required (which one represents 2^0, which one 2^1 and so on). In most digital computers this is done on a time basis, by referring the pulses to a continuously generated train of " clock pulses " which act as a time standard. Each of these clock pulses establishes a time interval which represents a particular numerical value—2^0, 2^1, 2^2, 2^3 and so on, as shown at the top of Fig. 8.3. Thus the value of a stored pulse arriving at the output of the register is determined by which of these clock-pulse intervals it occurs in. And this, of course, only works out properly if the numbers are fed into the store on the same time scale in the first place.

It will be noted from Fig. 8.3 that the clock pulses do not go on increasing in numerical value to infinity, but are repeated in regular cycles so that the first pulse in the second cycle goes back to value 2^0, and so on. Each of these cycles is the period in which a complete number is formed by a pulse train, and is therefore called a *number period*. In the actual computer it has to be made long enough to accommodate any number which may occur in a calculation. Usually this amounts to something like 30 to 40 digit pulses (corresponding to about eleven decimal places) but for the

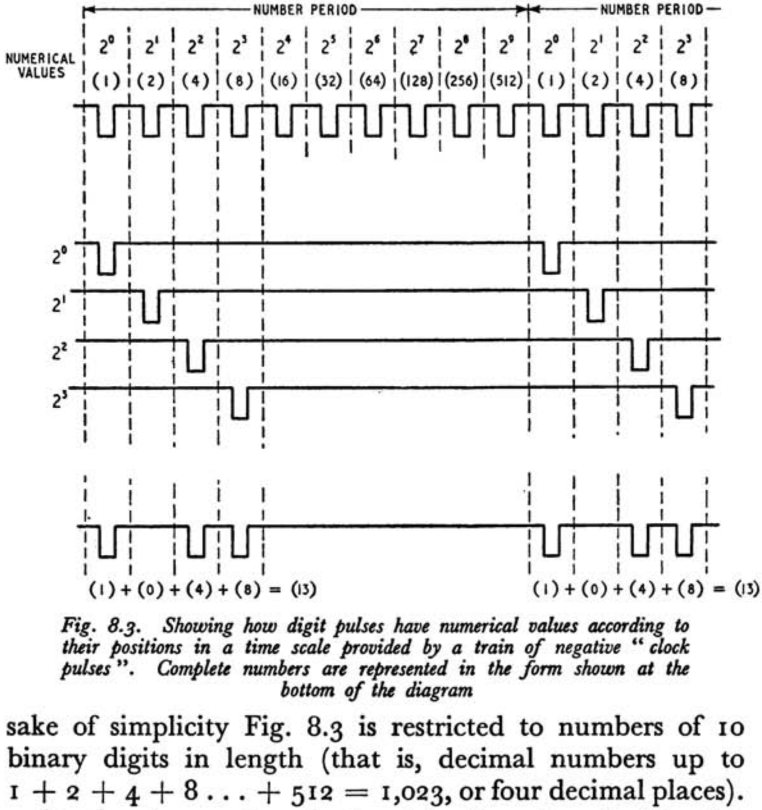


Fig. 8.3. Showing how digit pulses have numerical values according to their positions in a time scale provided by a train of negative "clock pulses". Complete numbers are represented in the form shown at the bottom of the diagram

sake of simplicity Fig. 8.3 is restricted to numbers of 10 binary digits in length (that is, decimal numbers up to $1 + 2 + 4 + 8 \ldots + 512 = 1{,}023$, or four decimal places).

As already mentioned, the delay-line store in Fig. 8.2 is arranged to have sufficient delay to accommodate one complete number, so in a computer using the Fig. 8.3 number period this would correspond to an interval of ten clock pulses. Thus if a number like 1101 is fed into the store in a particular number period, each digit pulse will appear at the output at the corresponding time in the next number period, as shown at the bottom of Fig. 8.3, and so on with each subsequent circulation of the number. To extract the number from the store, then, it is only necessary to use the clock pulses to open an electronic gate circuit connected to the output of the line. The particular intervals in which the digit pulses emerge during the number period will then identify their numerical values. A similar gating process under the control of the clock pulses is used for feeding the numbers into the store in the first place.

The clock pulses, however, are important not only to this particular type of storage register but to the operation of the whole computer. They provide a standard time reference by which every digit pulse in the machine is given a numerical significance. They are also used for synchronizing the control circuits by opening or closing switching gates at particular instants, and because of the cyclic nature of the number periods this gives a certain *rhythm* to the operation of the machine.

Each number period is analogous to a beat in music, with certain switching operations corresponding to the notes played on that beat. Just as a regular number of beats forms a bar of music, so the number periods are grouped into repetitive sequences which set the operational rhythm of the computer. This particular aspect will be dealt with in more detail later.

For the purpose of operating some of the gating circuits in the control section, clock pulses representing particular intervals in the number periods must be available. Thus, the 2^0 value pulses are required on one wire, the 2^1 pulses on another, the 2^2 on another and so on, as shown in the centre part of Fig. 8.3.

To produce these gating signals on different wires a kind of electronic distributor is used. The pulses from the clock-pulse generator are applied simultaneously to a series of " AND " gates, as shown in Fig. 8.4, the outputs of which form the ten wires. Each of the gates is then opened in turn (and closed again immediately afterwards), and this is done by using the pulse from each output, delayed by one pulse-period, to actuate the next gate, as shown. The gating signal therefore passes right

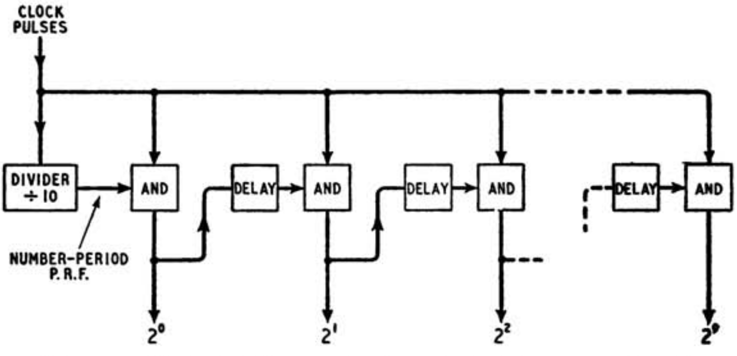


Fig. 8.4. Electronic distributor system for producing clock pulses on different wires for gating purposes

down the line of gates, distributing successive clock pulses on to successive output wires.

This process is repeated cyclically by initiating the whole train of events at a repetition frequency corresponding to the number period. The output from the clock-pulse generator is frequency divided by 10 as shown to give a series of initiating pulses at the number-period p.r.f., and these are applied to the first gate in the chain. Thus the clock-pulse allowed through this first gate becomes the first in the number period and takes the value 2^0 (that is, 1), while the others on the remaining nine wires become 2^1, 2^2, 2^3 . . . to 2^9.

The clock-pulses made available in this way can be used, for example, as a means of recording a number pulse-train in a static register of the kind shown in Fig. 8.1. This involves distributing the successive pulses in the train to the corresponding trigger circuits in the row, putting them into the " set " condition for a " 1 " digit and leaving them in the " unset " condition for a " 0 " digit. The number signal, consisting of negative digit pulses, is sent to the inputs of " AND " gates connected through diodes to all the left-hand grids of the trigger circuits, as shown. The digit pulses put the trigger circuits into the " 1 " state only if they coincide with corresponding clock pulses applied from the appropriate wires in Fig. 8.4 to the other inputs of the " AND " gates. If, for example, there is a pulse in the number signal at the time of the 2^1 clock pulse, trigger-circuit T_1 will be put into the " 1 " condition. To clear a stored number when it is no longer required, the trigger circuits are " unset " into the " 0 " condition by a pulse applied to all the right-hand grids.

We are now in a position to look at the general organization of the control circuits. These, of course, differ considerably from computer to computer, depending on what sort of application the machine is designed for. There is no standard arrangement for a digital computer, but the block schematic diagram (Fig. 8.5) contains most of the essential features found in a good many machines. It is purely a hypothetical computer, of course, and is greatly simplified for the purpose of explanation. The arithmetic circuits, for example, are represented by just an accumulating adder (centre) based on the delay-line register in Fig. 8.2.

It will be assumed that the numbers to be operated on are in the form of 10-digit binary numbers, each of which is stored in a delay-line register of the kind already described. The programme of instructions, listing the operations to be performed, is translated into a numerical code (by a method to be described in Chapter 11) and each of the instructions again consists of a 10-digit binary number held in a delay-line register. There are actually 32 of these registers available altogether for the numbers and instructions, and they are grouped together (as shown in the top part of Fig. 8.5) in a section called the *main store*. This has a common input line on the left and a common output line on the right.

In order to identify the registers (and hence the numbers in them), each one is given a serial number called its *address*. In Fig. 8.5 the addresses are 01, 02, 03 . . . to 32,

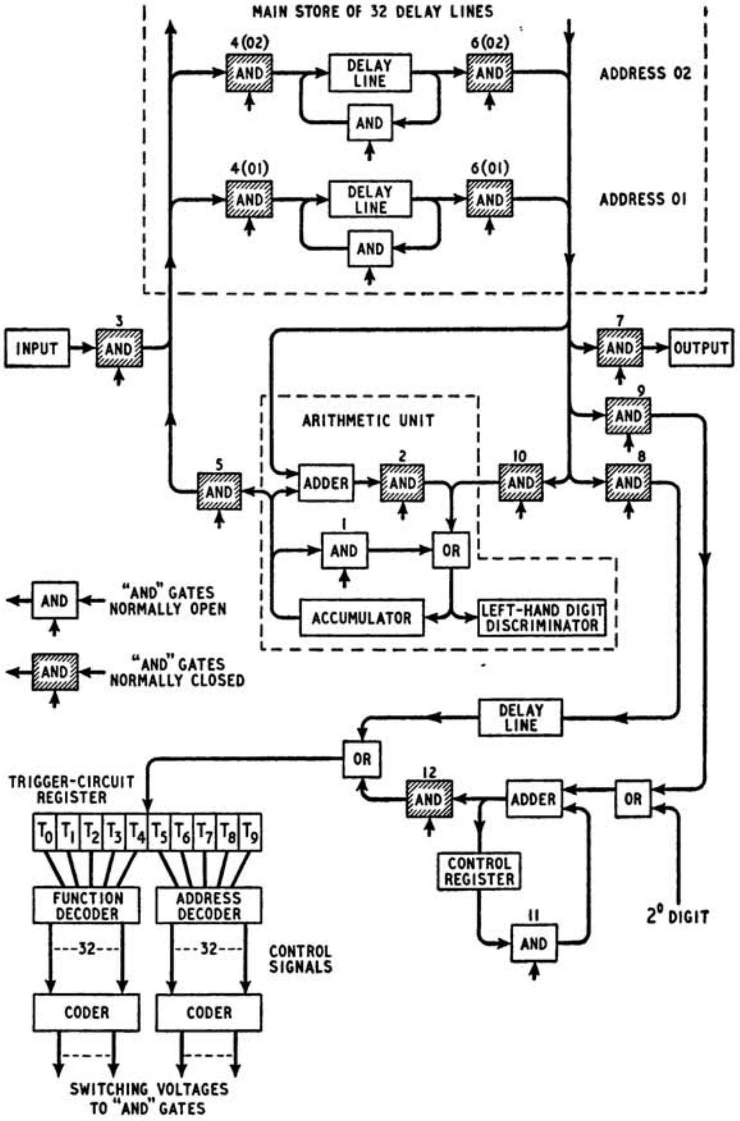


Fig. 8.5. Schematic of the control system of a hypothetical computer, showing how instructions are conveyed from the main store to coding circuits which produce control signals for switching purposes

and these tags, converted into binary notation, are actually used in the instructions for referring to the numbers held in the respective registers. For example, an instruction might call for some operation to be performed on " the number stored in register 07 ". Thus, one part of the 10-digit instruction contains the address of the number referred to, while the remaining part signifies, again in binary coded form, the *function* which is to be performed on that number. We shall assume that 5 binary digits are used for each part.

The first step in converting the instructions into switching signals is to transfer them to the trigger circuit register on the bottom left of Fig. 8.5—the actual distribution of digit pulses into the static form being done as was demonstrated in Fig. 8.4.

The " function " part of the instruction is sent to the five trigger circuits T_0 to T_4 and stored as a pattern of digits, while the " address " part is sent to the remaining five circuits T_5 to T_9 and stored as another digit pattern. Next, the outputs of the trigger circuits pass to two " decoding " circuits—the *function decoder* being fed from trigger circuits T_0 to T_4 and the *address decoder* from the circuits T_5 to T_9.

One of the decoding circuits is shown in Fig. 8.6. It consists of 32 5-input " AND " gates. Each of these gates takes one output from each of the five trigger-circuits (either the direct or the inverted output, see Fig. 8.1), there being 32 different ways of making the connections. Thus, for each possible combination of digits in the five trigger circuits, a particular " AND " gate will give a negative voltage on its output wire.

These output wires provide the control signals for switching the rest of the computer. Each one, however, may be required to operate a whole group of gates simultaneously —opening some and closing others—in order to put an instruction into effect. This means that the control signal must first be applied to a *coder*, which is rather like the

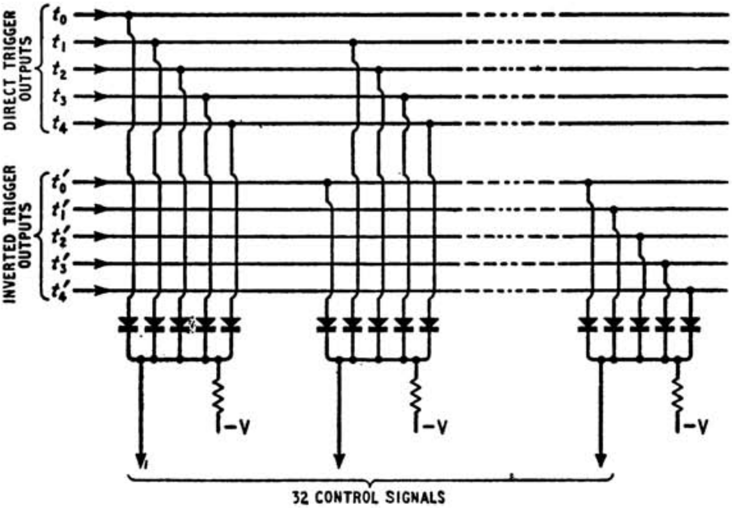


Fig. 8.6. Decoding circuit giving 32 control signals from different combinations of the outputs of 5 trigger-circuits in a register

Fig. 8.6 decoder working in reverse and providing a particular pattern of switching voltages on several wires for every input control signal. The coder output wires are connected permanently to the various " AND " gates in the control circuits but only operate them when the appropriate patterns of instruction digits arrive in the trigger-circuit register. Thus in order to make a particular coded instruction operate a particular group of gates one has to ensure that the right output wires from the coder are connected to these gates.

It may appear that the coder is to some extent reversing the action of the decoder and that consequently both of them could be dispensed with. This is partly true, and it is possible to use voltages obtained directly from the trigger-circuit register for switching the gates. The scheme, however, has the disadvantage that the " function " part of the instruction requires more digits and trigger circuits than are really necessary to provide the switching information. As a result more space is needed for the instructions in the main store. The use of a decoder and a coder, as in Fig. 8.5, generally offers greater range and flexibility for a limited number of instruction digits.

It now remains to be seen how the coded instructions are actually transferred from the main store to the trigger-circuit register. The instructions are actually held at successive addresses of the main store in the order in which they will be required. It is therefore necessary for these registers to be selected in turn and the numbers in them (which, of course, are circulating pulse trains) to be passed along the required channels by opening suitable gates.

The whole process is initiated by the " control register " at the bottom of Fig. 8.5. This holds, in binary number form, the address of the stored instruction which has just been obeyed by the computer. By means of an associated adder, the number " 1 " (obtained from the 2^0 clock pulse) can be added to it. Since the instructions are stored at successive numerical addresses, this process forms the address of the next instruction to be obeyed (e.g., address 06 plus 1 becomes address 07). The newly formed address is sent to the " address " part of the trigger-circuit register and the associated decoder produces a control signal which opens the output gate of the register containing the appropriate instruction. The instruction itself then passes to the trigger-circuit register, and both its " address " part and its " function " part are decoded to produce the desired operation.

All this, however, is done in an orderly sequence based on the rhythm of the machine. In our hypothetical computer it is a " four-beat-to-the-bar " rhythm—that is to say, the machine operates in regular cycles of four number periods These run as follows:—

(1) In the first period the address number stored in the control register is sent to the associated adding circuit

through " AND " gate 11, which is opened by a clock pulse. Here 1 is added, forming the address of the next instruction. The new address goes back to the control register ready for use in the next " bar " or cycle. It also goes to the " address " part of the trigger-circuit register via " AND " gate 12, which is opened by a clock pulse.

(2) In the second number period, the address in the trigger-circuit register is decoded by the address decoder. This causes one of the " AND " gates 6 (01, 02 . . . etc.) to be opened, and allows the instruction stored at the specified address to pass out of the main store. It actually passes through " AND " gate 8, which is opened by a clock pulse. It cannot, however, go straight into the trigger-circuit register because the " address " part of this is already being used for the selection of the instruction, so it is held back in a delay-line until the next number period.

(3) In the third period the instruction emerges from the delay line and passes through the " OR " gate into the trigger-circuit register, its own " address " part taking the place of the instruction-selecting address that was there in period (2).

(4) In the fourth number period the instruction is decoded by the address and function decoders, and these open or close gates which cause the arithmetic or other operation specified by the instruction to be carried out.

This cycle is repeated, taking a new instruction every time until all the required operations on the stored numbers have been performed. The last instruction in the programme is a binary number which signifies " stop the control cycle ". When this is transferred to the trigger-circuit register, the function decoder produces a switching signal which brings the whole system to a standstill.

A great many of the operations associated with the arithmetic processes are simply transfers of numbers from one register to another, and these have to be carried out in phase (4) of the control cycle as well as the arithmetic operations themselves. For example, in one cycle the decoders might cause a number held at a particular address to be sent to the accumulator. For this, gate 1 would be closed, gate 10 opened and one of the gates 6 opened (according to the signal from the address decoder). In another cycle, the result of an arithmetic process held in the accumulator might be transferred to the main store. Here gate 5 would be opened, and also one of the gates 4 (by the address decoder) and the corresponding gate 6 closed. Then there would be operations for transferring numbers from the input unit to the main store (opening gate 3 and one of 4 and closing the corresponding gate 6) and from the main store to the output unit (opening one of gates 6 and gate 7).

For the actual process of adding a number from the main store to the contents of the accumulator, gate 1 would be closed, gate 2 opened and one of 6 opened. Another instruction might cause a number from the main store (actually an address number) to be transferred to the control register. As a result the computer would not carry out the next instruction in the normal sequential manner, but the control system would " jump " either backwards or forwards in the programme to some other instruction, as specified by the address number. For this, gate 11 would be closed, gate 9 opened and one of gates 6 opened.

Such a " jump " in the normal control sequence might be put into operation automatically only when, say, the left-hand digit of the number in the accumulator was a " 1 ". This would enable the sequence of instructions to be modified according to the partial results obtained during the calculation—on the principle already mentioned in Chapters 1 and 2. The left-hand digit might, for example, be used as a sign to indicate whether the number in the accumulator was positive or negative. More will be said about the use of this " conditional operation " in Chapter 11. There are, of course, a great many other operations which would be carried out in a practical computer, but no doubt the reader has already reached some sort of understanding of how the control circuits do their work. The general organization of the machine can now be summarized by a smaller block diagram as shown in Fig. 8.7. Here the somewhat complex arrangements of Fig. 8.5 have been " boiled down " into five main functional units, which are self-explanatory, plus the clock-pulse generator for synchronisation. The " storage unit " for example, represents not only the main store but all the other registers which may be distributed about the machine. Similarly with the " control unit ". Although in fact the control system spreads into every part of the computer, like the nervous system in the human body, it is shown here for convenience as a distinct and separate unit.

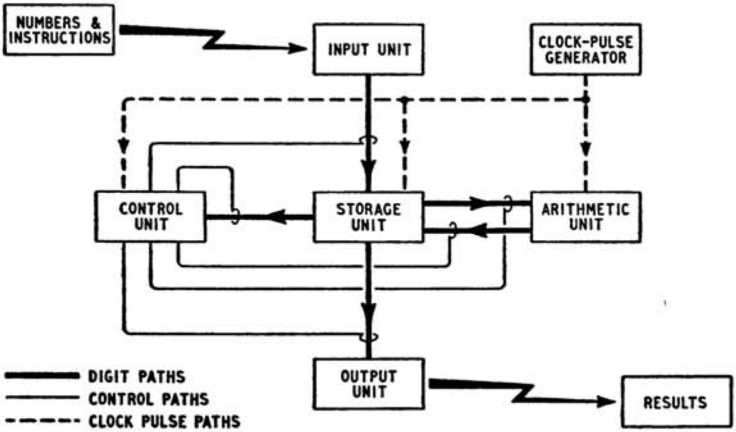


Fig. 8.7. Simplified block diagram of the complete hypothetical digital computer

As for the " input unit " and " output unit ", these are electromechanical devices for respectively reading-in and writing-out numerical information. The input and output media might be punched cards, punched paper tape or magnetic tape, and the associated equipment converts the numerical information from this form into electrical pulses and from pulses back again. More will be said in Chapter 10 about the actual equipment used. Incidentally, the input and output units are quite likely to contain temporary storage registers acting as " buffers ", to facilitate the transfer of numbers to and from the main store.

The hypothetical computer in Fig. 8.5 would actually be classed as a *single-address* machine. This refers to the fairly obvious fact that it is worked by instructions which contain only one address as well as the " function " part. The term also implies that the instructions are obeyed in the order in which they are put into the main store (by the process of adding " 1s " in the control register). There are, however, *two-address*, *three-address* and even *four-address* machines. Here the extra addresses in the instructions may be used for specifying the storage locations of other numbers to be included in the operations. Sometimes, however, one of them is used to specify the serial number of the next instruction to be obeyed—and this means, of course, that there is no need to select the instructions in the exact order in which they have been put into the machine. Each instruction gives the address of the next one. This facility is sometimes utilized in what is known as " optimum programming ", for increasing the operating speed of the machine when the main store is of a kind which introduces a certain amount of delay in producing the numbers. A great many digital computers, however, are based on the straightforward single-address system with instructions taken in natural sequence, and Chapter 11 on programming will be mainly concerned with this mode of operation.

9

STORAGE SYSTEMS

THE last two chapters have already shown the great importance of storage systems in digital computers. The " mechanized memory " is, in fact, the main thing which distinguishes the automatic computer from the manually operated calculator. It performs most of the functions for which a human operator is necessary on the simpler machine —supplying numbers, storing partial results, holding the programme of instructions and so on. In practice the whole design of a digital computer is influenced by the characteristics of whatever storage system is chosen, so it is not surprising that " memory " devices should have become a complete subject of their own. Moreover, digital computers are now being adapted for use in offices as mechanized filing systems, and here, of course, the " memory " unit plays the most important role of all, while the other circuits are merely used for ancillary operations, such as transfer of numbers and accumulation of running totals.

The two most desirable features of a storage system are (*a*) that it should have as large a storage capacity as possible for a given physical size and manufacturing cost (or, alternatively, be as small and inexpensive as possible for a given storage capacity) and (*b*) that it should make the stored information available with the minimum of delay. The first requirement makes for small and low-cost computing equipment while the second makes for high-speed operation. Unfortunately, these two requirements often conflict with each other. For example, in the static register using a row of valve circuits described in the last chapter the stored binary digits are available immediately and simultaneously as the electrical conditions of the output wires. But the system demands at least one thermionic valve (and probably a double valve) for every digit stored. A large-capacity store based on this technique would be far too bulky and expensive. On the other hand, a length of magnetic tape has an enormous capacity for its size and cost, storing a hundred or more binary digits to the inch; but the information can only be made available by the comparatively slow process of moving the tape past a reading head.

The digital computer, then, has to be designed around these limitations, using different types of storage devices in different parts of the machine. Stores which are used frequently in calculations have to be " quick-access " types to obtain high-speed operation, but because they are often large and expensive in relation to their storage capacity they have to be restricted in number. Commercial computers use anything from about 15 to 200 registers, depending on the size of the machine, for such high-speed work. Normally this would be something of a limitation on the capabilities of the computer. In practice it is overcome by " backing-up " the high-speed section with a reserve store of large capacity—of the type in which high density of storage is obtained with reasonable size and cost, like magnetic tape—and transferring blocks of numbers to the high-speed section when required. This large-capacity store is usually a " slow-access " type, but this does not matter greatly as it is not used very frequently. Depending on the principle employed, such " backing-up " stores have capacities of anything from about 5,000 numbers upwards (or about 150,000 binary digits up to several million).

Another kind of distinction between storage systems is made according to the way in which the numerical information is held in them. Some, like the row of valve or transistor circuits, are called *static* stores because the binary digits are laid out in a stationary pattern. Others, like the delay line system, described in the last chapter, are called *dynamic* stores because the information is continually on the move. And these dynamic stores can be divided again into (*a*) types in which the digit pulses are circulating within the storage medium, and (*b*) types in which the medium itself is mechanically circulating. Yet another distinction can be made between stores in which the information is erased by the process of reading out and those in which it is retained.

Dealing first with static storage systems, the reader will have gathered already from Chapter 8 how a row of two-state trigger circuits can be arranged to provide such a store. This principle, of course, is not confined to thermionic valves. Transistors, cold-cathode tubes, electromechanical relays and any other devices which will give two-state operation under electrical energization can be used in a similar way.

In the last few years a great many new and experimental two-state devices have been developed, based on magnetic,

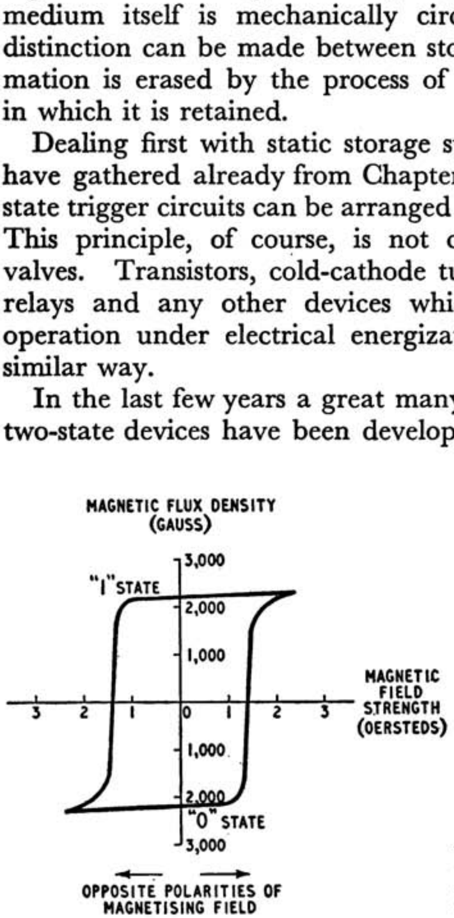


Fig. 9.1. Rectangular hysteresis loop of ferrite magnetic material, showing the two states of magnetization for binary storage systems

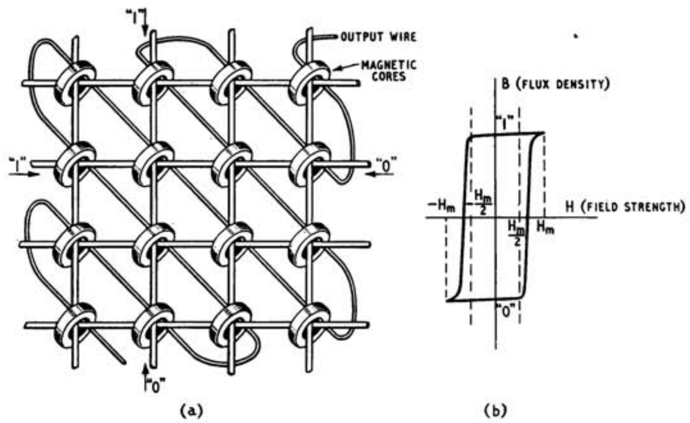


Fig. 9.2—above. Arrangement of magnetic cores in a matrix storage system (a), with (b) rectangular hysteresis loop of the core material. Plate 15—below. One plane of a 32 × 32 magnetic matrix store made by Mullard

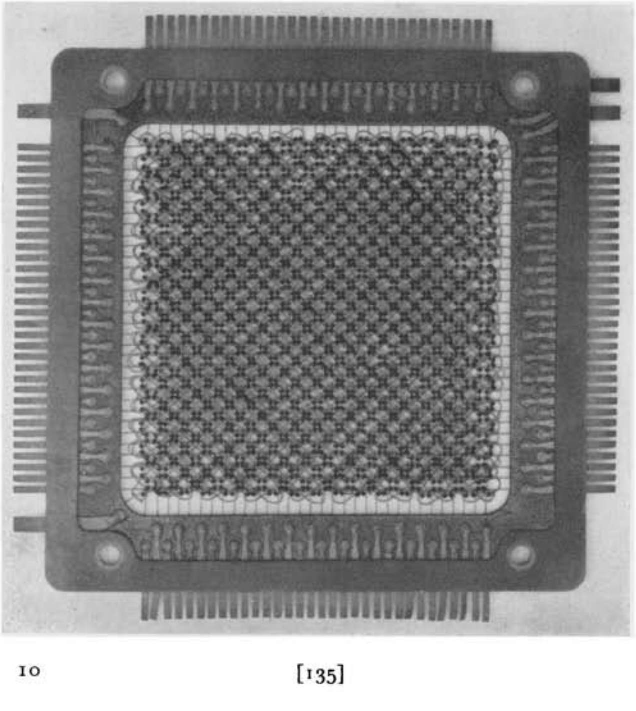


Plate 16 (a)—below. Ferro-electric store constructed as a matrix on a single crystal of barium titanate

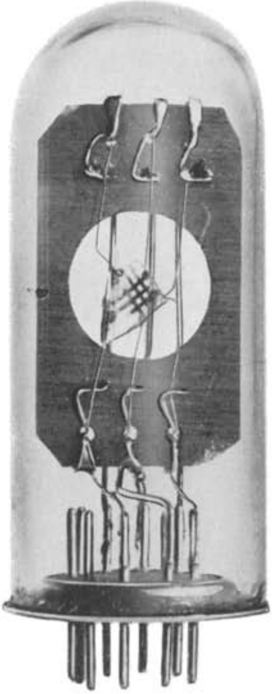
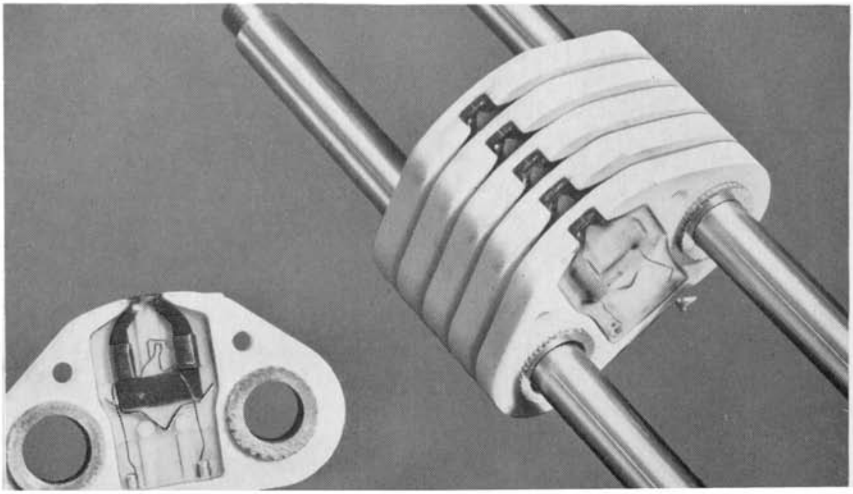


Plate 16 (b)—above. Stack of 40 magnetic matrix planes (Plate 15) giving a storage capacity of 163,840 digits

Plate 16 (c)—below. Construction and mounting of read/record heads for a magnetic-drum store



dielectric, semiconductor and superconductivity phenomena. Some of these are described in Chapter 13.

One of the new devices which has passed the experimental stage and is now being used extensively in digital computers is the two-state magnetic core. The two states in question are actually conditions of high remanent flux density (almost at saturation point) in the core. These are obtained by virtue of the fact that the core material—usually a ferrite—exhibits an almost rectangular hysteresis loop on its graph of magnetic flux density against magnetic field strength instead of the more usual gently curving loop.

Fig. 9.1 is an example obtained from a ferrite core. Suppose the core is magnetized to saturation in one direction, say to the " 1 " state, by passing a pulse of current through a winding. At the end of the current pulse the magnetization will still be very high—at least 90 per cent of the maximum flux density—thereby retaining the information that the core has been switched to the " 1 " state. If now another current pulse is passed through another winding so as to apply a magnetizing field of opposite polarity, the magnetic flux will at first decrease slowly, then upon reaching the top-left " knee " will suddenly drop to zero and shoot straight up to the opposite saturation point or " 0 " state. Thus a fairly small increase of magnetizing current will switch the core rapidly from one state to the other. Once the core is in that state it is not critically dependent on the current, which can be reduced to zero without affecting the stored information.

A row of these cores with suitable windings can be set up to form a storage register in the same way as a row of valve or transistor trigger circuits. The principle is used most extensively, however, in what are known as magnetic matrix stores. The matrix consists of rows of horizontal and vertical wires, as shown in Fig. 9.2 (a), with tiny ferrite cores threaded on to them so as to enclose all the intersections. An actual example, with 32 horizontal and 32 vertical wires, is shown in Plate 15, p. 135. Each of the wires forms a one-turn magnetizing winding where it passes through a core, and the idea is that a core is only switched from one state to the other when there is a coincidence of currents through the horizontal wire and the vertical wire enclosed by it. The two currents provide a total magnetic field, shown by H_m in Fig. 9.2 (b), which is sufficient to drive the core to the other state, but a current in only one wire gives only half the necessary field, shown by $H_m/2$, and therefore has no effect.

Thus, it is possible to store a binary digit, 0 or 1, on any of the cores by applying current pulses to the two wires which interesect at that point. Whether a 0 or a 1 is stored depends on the direction of the current, as indicated in Fig. 9.2 (a). The fact that each wire is common to a good many cores obviously makes for a more compact and convenient type of storage than if, say, each core had its own individual energizing winding. In the Plate 15 matrix, for example, there are $32 \times 32 = 1024$ cores, capable of storing that number of binary digits. By orthodox methods this would entail 1024 energizing inputs, but on the matrix principle it only requires $32 + 32 = 64$ inputs.

Reading out of the stored information is also achieved by applying current pulses to the appropriate pairs of wires. If a particular core is in the " 1 " state the application of coincident read-out pulses of, say, " 0 " direction to its two intersecting wires causes the core to be switched over. The resultant large change of flux induces a voltage pulse in the output wire shown in Fig. 9.2 (a) which threads all the cores. The presence of this output pulse indicates that a " 1 " was stored. If, on the other hand, the core had been in the " 0 " state, the application of the " 0 " read-out pulse would have made no real difference to it. There would have been no large change of flux and no voltage pulse induced in the output wire—thereby indicating that a " 0 " was stored.

Using ferrite cores of only 2 mm in diameter, the storage capacity is quite large for a given size, and in practice numbers of the matrix planes are stacked up vertically, as shown in Plate 16 (b), to give even greater capacity. And, of course, this is a static type of store giving high-speed access to the information. The main disadvantage is that rather heavy currents are required through the wires to switch the cores, and this means that the incoming digit pulses have to be passed first through special driving amplifiers.

Another type of store based on a rectangular hysteresis loop—but this time in a dielectric material—is the ferroelectric cell. In this an electric field is applied to a ferroelectric material such as barium titanate.

The graph of the resultant electric flux density plotted against electric field strength is a rectangular hysteresis loop similar to the magnetic one, as shown in Fig. 9.3 (a).

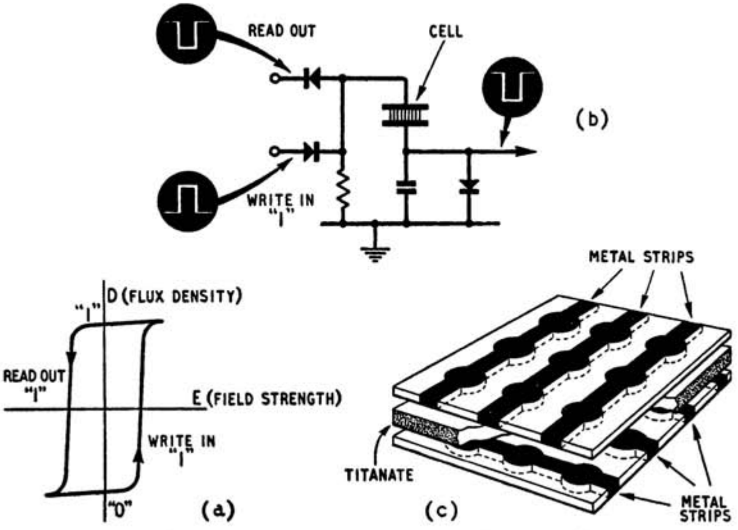


Fig. 9.3. Ferroelectric storage system based on the hysteresis characteristic of a titanate dielectric. The hysteresis loop is shown at (a); a capacitor type of storage cell at (b); and a matrix incorporating a large number of such cells at (c)

A cell to store one digit is made in the form of a capacitor with two metal electrodes on either side of a piece of ferroelectric material acting as the dielectric. When a voltage pulse, say positive, is applied to the cell by the arrangement shown in Fig. 9.3 (b) the capacitor is left with a positive remanent induced charge, indicating that, say, a " 1 " digit is stored, as in (a). To read out the information, a negative pulse is applied to the cell by the other terminal in (b). When a " 1 " is stored, the reversal of polarity caused by the read-out pulse produces a negative pulse at the output, whereas if a " 0 " had been stored the read-out pulse would have produced no change and therefore no output pulse.

As with the magnetic store, the ferroelectric store can be arranged in the form of a matrix, with a sheet of the titanate dielectric between two arrays of metal electrodes, as shown in Fig. 9.3 (c). Application of a voltage pulse to one X electrode and one Y electrode induces a charge at their point of intersection and so stores a digit there. Plate 16 (a) shows a small experimental matrix of this kind mounted on a single crystal of barium titanate. In general the ferroelectric store has the advantages of smaller size and lower power consumption over the two-state magnetic store, but requires higher operating voltages and is somewhat sensitive to environmental changes. So far it has not been used very widely in digital computers.

Incidentally, the ordinary capacitor with a straightforward dielectric provides a simple but quite effective means of storage by means of its ability to hold an electric charge. Matrix stores have been constructed on this principle, but these again are not used very extensively.

Both the magnetic and ferroelectric systems have the disadvantage that the stored information is erased upon reading out so that a continuous indication of the contents of the store cannot be obtained. A device which overcomes this disadvantage is the ferroresonant store. Ferroresonance is a condition which occurs when a saturable inductor and a capacitor resonate at a certain frequency of applied alternating voltage. The increased current at resonance keeps the inductor core saturated and the inductance down to the required value for resonance. For a given frequency the inductor has two possible stable states—a high-current, low-inductance state, at which it resonates with the capacitor, and a low-current, high-inductance state at which no resonance occurs.

The digit pulse to be stored is applied through a trigger winding coupled to the main inductor, and the resultant magnetic flux causes the inductor to be put into the ferroresonant condition. Unfortunately it cannot be switched back out of resonance by another pulse through the same trigger winding. In practice, a two-state store consists of two ferroresonant circuits with trigger windings, arranged so that triggering one circuit into resonance puts the other out of resonance. This is comparable with the conducting and non-conducting conditions of the two valves in the Eccles-Jordan trigger circuit, and makes available two input terminals (for " set " and " unset " pulses) and two output terminals (for direct and inverted outputs). Additional advantages of the system are low power consumption, cheapness, reliability and fast operation, though of course a special supply of alternating current is required.

Coming now to dynamic storage systems, probably the easiest one to understand is magnetic tape storage because the equipment is basically similar to the familiar domestic tape recorder. In some ways the digital tape storage system is simpler, for considerations like distortionless reproduction of the recorded information do not matter very much when one is only concerned with binary on-off signals. On the other hand, the mechanism usually has to run at high tape speeds—say 100 in/sec—and be capable of rapid stopping, starting and reversing to reduce the access time when selecting blocks of numbers for transference to high-speed stores. The recording is done by applying the digit pulses to the energizing coil of a recording head with a very narrow gap in its pole-pieces, across which is drawn the magnetically coated tape, as shown in Fig. 9.4. The pole-pieces of the head are arranged in relation to the tape so that the small magnetic dipoles induced in the coating have their flux lines running along the length of the tape, not across it.

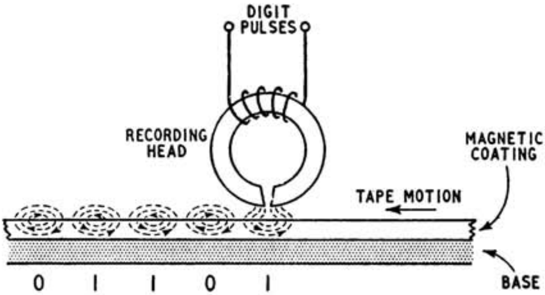


Fig. 9.4. Principle of storing digital data on magnetic tape. The direction of magnetization depends on whether a 0 or a 1 is stored

This makes it possible for a number of parallel tracks to be recorded without mutual interference, and in fact most data recording heads are constructed for this multiple-track operation (see p. 154, Plate 18 [a]).

In one system digit pulses are applied to the head as an a.c. waveform, as shown in Fig. 9.5 (a), so that the "1" pulses produce currents in one direction through the energizing coil and the "0" blank spaces produce currents in the opposite direction. Each flow of current magnetizes the tape to saturation and sets up a small dipole or cell of magnetization, the polarity of which depends on the digit stored, as shown in Fig. 9.4. The waveform in Fig. 9.5 (a) can be considered as providing a continuous flow of current through the head in one direction (above the datum line) which is interrupted at intervals by pulses of current in the other direction representing the "1" digits. On the tape this gives a uniform magnetization in one direction (the " 0 " direction) which is broken by small magnetic dipoles of the reverse direction (the " 1 " direction).

Reading out is, of course, done by drawing the magnetized tape across the pole-pieces of a similar head (or the same one). The coil then gives out voltage pulses in response to the changes in the flux linking the pole-pieces. Since the read-out head responds to the *changes* of flux occurring between the small fields in Fig. 9.4, it produces voltage pulses as in Fig. 9.5 (b) corresponding to the leading and trailing edges of the original digit pulses at (a). The (b) waveform therefore has to be converted back into the form of normal digit pulses by using, say, the positive excursions to trigger or gate a source of clock pulses.

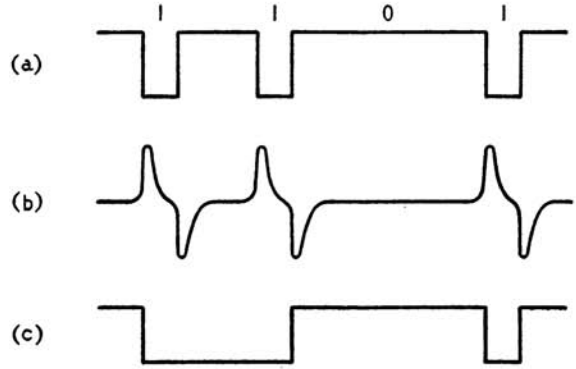


Fig. 9.5. Train of digit pulses applied to a recording head is shown at (a); the waveform obtained from the read-out head at (b); and the waveform used in the " non-return-to-zero " system of recording for representing the number in (a) at (c)

This system of recording is known as the "return-to-zero " system because there is always a boundary between the digits impressed on the tape. In another system called " non-return-to-zero ", wherever a sequence of consecutive " 1s " occurs in the number the recording-head current remains constant in one direction and whenever a sequence of consecutive " os " occurs it remains constant in the other direction. This is illustrated in Fig. 9.5 (c) with the joining together of the two adjacent " 1 " pulses into one long pulse. The current only changes in direction when a " 0 " follows a " 1 " or a " 1 " follows a " 0 ". One advantage of the system is that it makes possible a closer spacing of the digits along the tape because there are no reversals of polarity between similar consecutive digits.

In a typical commercial magnetic-tape data storage system (Plate 19 [a], p. 171) a 2,400 ft. spool of tape will store 1½ million binary digits per track at a density of 100 pulses to the inch. The tape will accommodate eight parallel recording tracks on a ½-in. width (or four tracks on ¼ in. width) and runs at a speed of 100 in./sec. The drive mechanism (a subject which will be discussed more in the next chapter) permits starting, stopping or reversing in less than 10 milliseconds.

Similar principles of recording are used in the magnetic drum storage system. This type of store is widely used in commercial digital computers and has been found most reliable. It has a large storage capacity (several hundred thousand digits up to about a million) for a reasonably compact size, the information is not destroyed on reading-out and the access time is never any longer than the period of one revolution of the drum. The equipment usually consists of a non-magnetic drum coated with a magnetic material and an array of recording/reading heads arranged to give several hundred parallel tracks round the periphery as the drum rotates. The speed of rotation is normally in the region of 3,000 to 6,000 r.p.m., but some equipments have been made with speeds as high as 75,000 r.p.m. and others as low as 180 r.p.m.

The return-to-zero method of recording is illustrated (Fig. 9.6), and it will be seen that small magnetic dipoles are impressed on the magnetic coating as it rotates to form storage cells. As with magnetic tape, the flux lines run

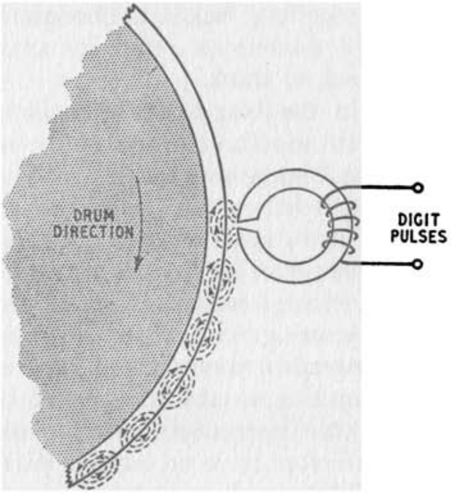


Fig. 9.6. Principle of magnetic recording for storage on the surface of a rotating drum

longitudinally along the track and the polarity of magnetization in each cell indicates whether a "1" or a "0" is stored. On reading out, the waveform obtained from the head is ideally like Fig. 9.5 (b), but in practice it often degenerates into something like a triangular or sine wave with peaks corresponding to the leading and trailing edges of the original recorded digit pulses. (This also applies in practice to magnetic tape storage.) To recover the original rectangular pulses, a common procedure is to limit and square the deformed wave and then sample it with very narrow gate or "strobe" pulses synchronized with the computer's clock pulses. Each strobe pulse establishes whether the particular excursion of the reproduced waveform occurring during that digit pulse period represents a "0" or a "1".

In order to record or read a number at a particular address on the drum (corresponding to one of the delay-line registers in Fig. 8.5 of Chapter 8) it is necessary first of all to select the appropriate track. This is normally done by switching the number pulses to a particular one of many fixed recording heads, although in some machines the heads themselves (a relatively small number) are moved from track to track.

Within the length of the track (i.e., the circumference of the drum) there is normally room for a good many complete binary numbers (say about 50) so the track can be divided into short sections, each corresponding to one register as in the main store in Fig. 8.5. Just as the Fig. 8.5 registers can be identified by address symbols, so can the angular storage locations around the drum tracks. This is done by using an extra control track on the drum and recording address symbols on it at regular angular intervals corresponding to the division of the track into registers. Programme instructions relating to such a storage system must therefore have an address part which specifies in code form (*a*) the track to be selected, and (*b*) the register to be selected within the track.

More details of the construction of magnetic drums will be given in Chapter 10. Their general appearance, however, can be seen from Plates 17 (a) and (b), p. 153, while Plate 16 (c), p. 136, shows the construction and mounting of one type of read/record head.

One of the great difficulties with storage devices for mechanized filing systems is in obtaining reasonably short access time when the store necessarily has a very large capacity. Moreover, it may be desirable to have " random access " to any piece of information—that is, without the necessity for running through the entire store to get to it. Magnetic drums have quite short access time but their capacity is considered low for this type of work (when compared with, say, magnetic tape) and it is necessary to use whole batteries of them in a single equipment. Magnetic tape, on the other hand, while capable of storing bits of information by the million, will certainly not give either short or random access to a particular item—as every owner of a tape recorder well knows. Here it is necessary to reorganize the clerical methods in such a way that the information is filed and extracted in a sequential fashion in large batches.

There is, however, one form of random-access magnetic memory which can overcome these difficulties. It looks rather like an automatic record player, being made up of a stack of 50 magnetic discs mounted on a vertical shaft and slightly separated from one another (see Plate 18 [b], p. 154). The stack rotates at 1,200 r.p.m. Information is stored as magnetized spots in concentric tracks on both sides of the discs, and at the side of the stack is a read/record device (pointed to by the man in the picture) which is moved automatically to the address of the required data. The storage capacity is 5 million binary digits, while the access time is of the order of milliseconds.

Another important type of dynamic storage system is the delay-line store already described in Chapter 8. In this, as distinct from the magnetic drums and discs, the storage medium is fixed and the digit pulses are circulating. The magnetostriction method of propagating the acoustic wave already mentioned is used quite extensively, and has the merits of simplicity, cheapness and reliability. A magnetostriction delay line constructed in the form of a standardized plug-in unit is shown in Plate 22 (b), p. 174. This is a register which will store one complete number of 42 binary digits with a pulse period of 3 microseconds.

The pulse period is significant here because it determines the actual capacity of the store. A given delay line will obviously hold twice as many digit pulses at 3-μsec intervals than it would at 6-μsec intervals. Storage capacity is also determined, of course, by the physical length of the delay line—more precisely by the distance between the sending and receiving coils on the line.

This distance has to be carefully adjusted so that, in the case of a register, the delay corresponds exactly to a number period (see Fig. 8.3). If this were not done, a digit pulse of a particular value, say 2^3, would not emerge from the line at the corresponding 2^3 interval in the next number period but slightly earlier or later. In subsequent circulations this error would increase cumulatively and eventually the original pulse would emerge as a 2^2 or a 2^4 value pulse —or even worse—in the number period concerned. Thus, when the complete number came to be read from the register the original pulses would be displaced from their correct time positions, relative to the number period, and it would be impossible to identify their true numerical values.

In practice it is not possible to adjust the line mechanically to the absolute degree of accuracy required, so the difficulty is overcome by avoiding the cumulative error which occurs with re-circulation. This is done by not circulating the digit pulses themselves but circulating clock pulses instead and using the digit pulses to gate them into the line according to the digit pulse pattern. And at each subsequent circulation the pulses and spaces emerging from the output of the line (with slight time errors) are used to gate fresh and accurately timed clock pulses into the input in the same digit pattern. Thus the slight error resulting from one traverse of the line is not allowed to build up, because the process starts afresh with new clock pulses in every cycle.

Other delay-line stores have used mercury columns or tanks for propagating the acoustic waves (notably in the ACE, DEUCE, LEO and UNIVAC digital computers) and here the waves are launched and picked up by quartz-crystal or other electro-mechanical transducers. Delays ranging from a few microseconds to about a millisecond are possible. The lines have the disadvantage of temperature sensitivity (which, however, can be overcome by thermal insulation) and have to be precision engineered. Rather more valves are required in the ancillary electronic circuits than with magnetostriction delay lines. In addition to columns, tanks and wires, it is possible to use solid blocks of material giving multiple internal reflections so that the acoustic wave path is " folded up " into quite a small space. One experimental 990-μsec delay line of this kind utilizes 30 reflections to fold up some 12 ft of wave path into a polygon $5\frac{1}{2}$ in. in diameter.

It will be recalled from Chapters 7 and 8 that delay lines are used not only for storage purposes in digital computers but also for the more direct purpose of delaying digit pulses. In Fig. 7.7 (d) of Chapter 7, for example a short delay of one digit period is required for holding the carry digit until the next higher value pair of digits arrives. For this sort of purpose the delay required may be only a few microseconds and it becomes possible to use electrical delay lines.

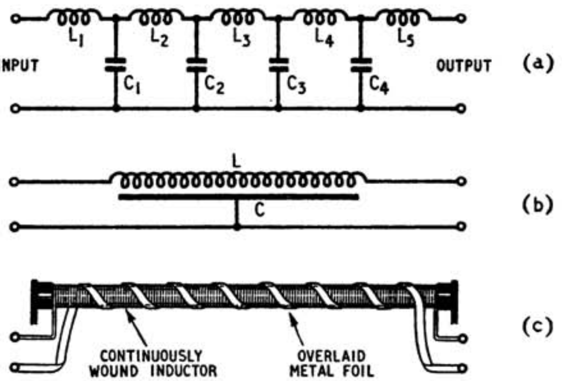


Fig. 9.7. Electrical delay lines: (a) lumped-constant circuit; (b) distributed-constant circuit, with a practical arrangement at (c)

One type, shown in Fig. 9.7 (a), is called a " lumped-constant " line because it is made up of separate inductors and capacitors—as distinct from the " distributed-constant " line in (b). The delaying action depends on the current phase lag (with respect to the voltage) in the inductors and the voltage phase lag (with respect to the current) in the capacitors. A voltage pulse applied to the input causes a current to flow into the first capacitor C_1. This current lags behind the voltage because of the back e.m.f. of the first inductor L_1. The capacitor proceeds to charge but the voltage across it takes some time to build up, so there is another lag—now with respect to the charging current, which is already delayed with respect to the input voltage. The same process occurs in the next inductor and capacitor, L_2 and C_2, producing an additional delay, and so on throughout the line until the input pulse eventually reaches the output.

In Fig. 9.7 (b) the inductance and capacitance are distributed over the whole length of the line, but in transmitting an input pulse the arrangement behaves as if it consisted of a large number of separate L–C sections as shown in (a). This is also true of coaxial cable and other commercially-available transmission lines, which are, in fact, occasionally used for delay purposes. A practical form of the distributed delay line is shown at (c)—the capacitance existing between the inductor wire and the overlaid metal foil.

Apart from these delay lines, the capacitor figures mainly as a device for static storage, as mentioned earlier. There is, however, a development of the straightforward electrostatic storage principle in a system which might be considered as being partly dynamic in operation. This is the cathode-ray tube store—a device around which the Manchester University computers were mainly designed but which has now been largely superseded for high-speed storage by the magnetic-core matrix.

An almost conventional cathode-ray tube is used, with its beam deflected to trace out a pattern on the screen comparable with a television raster. The beam is modulated on its grid to write " dots " for 0 digits and " dashes " for 1 digits on the fluorescent screen, as shown in Fig. 9.8 (a), and the screen becomes electrostatically charged at these points in a rather complex manner which depends on secondary emission. The visible screen pattern is shown at (b) with the potential distribution underneath, from which it can be seen whether a 0 or a 1 is stored.

Reading is accomplished by scanning the pattern with an unmodulated beam, and the arrival of electrons at each digit position causes a change of potential which depends, in its form and extent, on whether a dot or a dash is stored. This change of potential induces a voltage in a metal collector electrode which is placed on the face of the c.r. tube so that it forms a capacitor, with the fluorescent screen as the other plate. The resultant waveform on this collector electrode is shown in (b). Only the initial parts

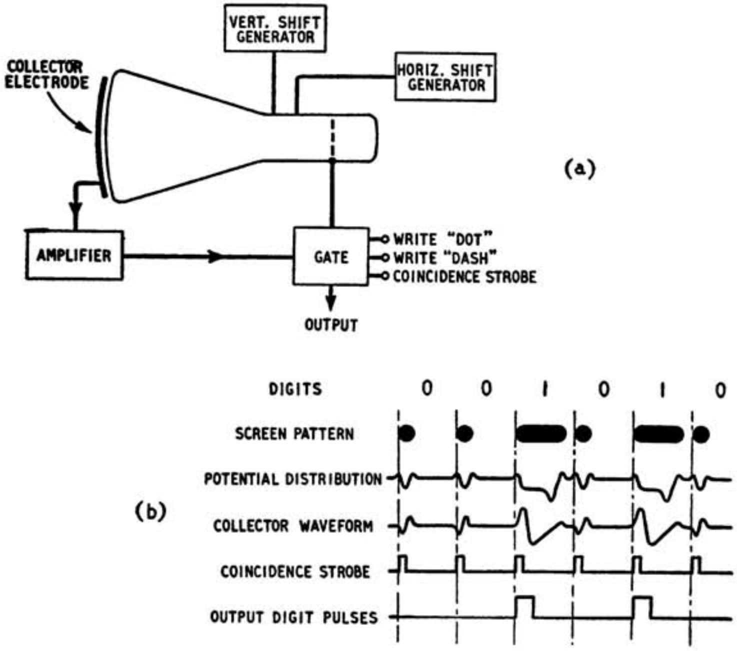


Fig. 9.8. Electrostatic storage system (a) based on a cathode-ray tube, and (b) the associated waveforms representing the binary digits

of the pulsations in the waveform are used to identify the digits (negative excursion for a 0, positive excursion for a 1), so the waveform is sampled by a coincidence strobe pulse at these points. The result of this process—whether there is coincidence or no coincidence—then gives an output pulse for a 1 and no pulse for a 0 as shown.

Since the charge pattern on the fluorescent screen tends to leak away over a period of time, it is necessary to regenerate it at regular intervals. This is done as part of the reading process. Actually the reading beam is not on continuously but is switched on by a pulse train at the particular moments when it crosses the digit positions on the screen. The "on" period of the reading beam corresponds to the duration of a dot, so it regenerates the 0 digits automatically. As for the 1 digits, when a pulse appears at the output, indicating the presence of a dash on the screen, it is used immediately to lengthen the "on" period of the reading beam and so re-write a dash on top of the original dash.

A typical cathode-ray tube store has 32 horizontal lines, each of which forms a register capable of storing a complete number of 40 binary digits in length. Other patterns besides the dots and dashes have been used, such as focused and defocused spots. The system as a whole, however, is not being used in modern computers and is only mentioned here in view of its historical importance and technical interest.

Large-scale digital computers often use a combination of the systems described above because of their requirements for large-capacity, slow-access stores in some places and quick-access low-capacity stores in others. The smaller and medium-sized machines generally compromise with a single storage unit, like a magnetic drum, which has medium access time and medium-to-large capacity and can be used for various purposes.

In general, magnetic drums, magnetic-core matrix stores and acoustic delay lines are probably the most

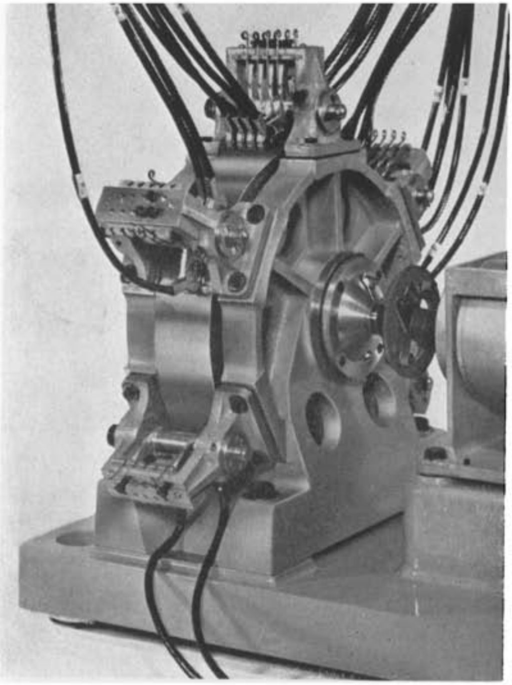
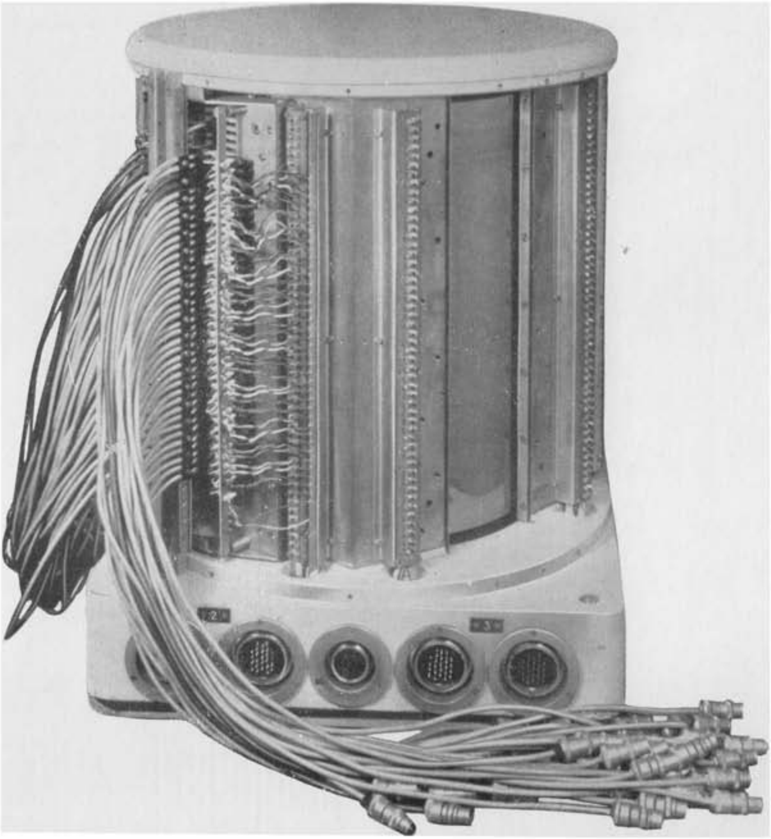


Plate 17 (a). Magnetic-drum store with horizontal bearings from a large digital computer. The read/record heads can be seen arranged at intervals around the periphery

Plate 17 (b)—below. This magnetic-drum store rotates about a vertical axis and is part of the equipment of another large digital machine



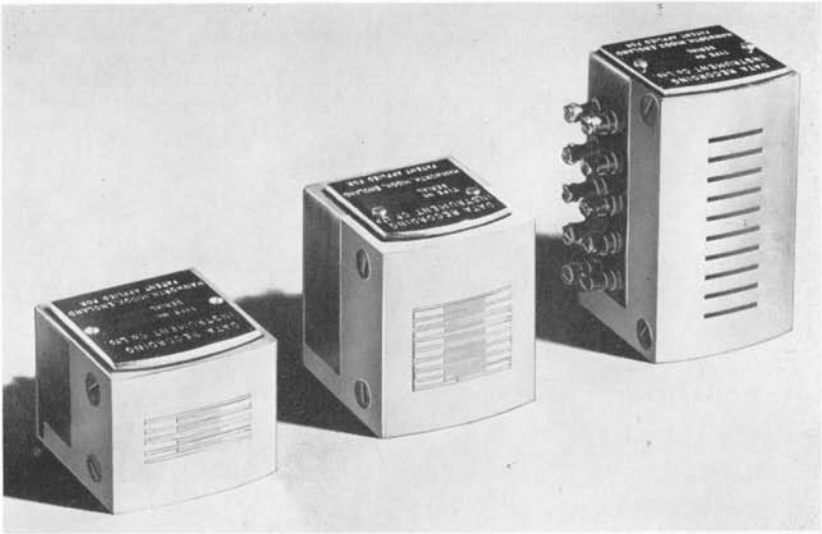
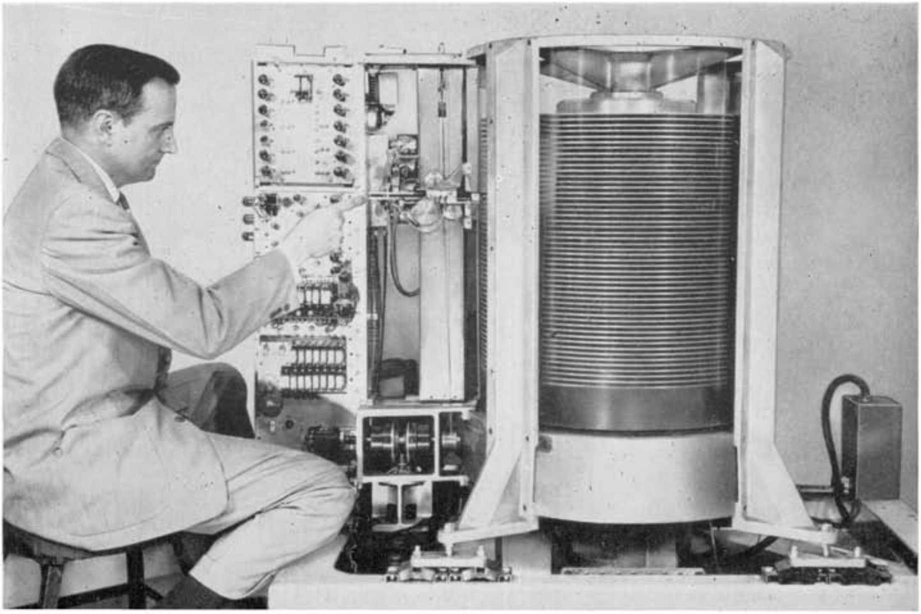


Plate 18 (a). Read/write heads with different numbers of pole pieces for multiple-track operation in magnetic tape storage systems

Plate 18 (b). Random-access storage system using magnetic discs. The man is pointing to the mechanism which carries the recording and reading head



widely used devices at present. There are also the various methods of storing the input and output data, before and after the actual computation. Here punched cards and punched paper tapes are the commonest media, although magnetic tape is now being used a great deal—sometimes directly and sometimes as " buffer " storage between the computer and orthodox punched-card or -tape equipment. But more will be said about input and output systems in the next chapter.

10

DIGITAL COMPUTER EQUIPMENT

HAVING now looked at the main functional units of the digital computer, one has some idea of what to expect in the " hardware " of a typical machine. Broadly speaking, one can take the block diagram Fig. 8.5 in Chapter 8 and say that each unit there is represented by a corresponding unit in the actual equipment. Of course, the mechanical design of the machine does not necessarily follow the pattern of the functional diagram and, again, the various manufacturers have their own favourite methods of construction. Nevertheless the block diagram is quite a helpful guide in finding one's way around the computer, and this is particularly true nowadays when the " unit construction " principle has become so common in the design of electronic apparatus. Here, each chassis or sub-chassis is not just a convenient receptacle for stowing away some of the electronic circuitry but is designed to perform a particular function in the operation of the complete apparatus.

It is not easy, of course, to decide just what is a " typical " digital computer, for nowadays the range of types available is extremely wide and diversified. There are special-purpose machines for doing only one particular job, such as controlling an industrial process, and general-purpose machines for handling any type of work. There are computers designed mainly for business data processing which can also handle scientific calculations, and machines intended for scientific work which can also undertake business computations. In terms of size, digital machines range from small transistorized equipments built into aircraft control systems (see, for example, p. 203, Plate 27 [a]) to complete rooms full of equipment for large-scale data processing (see p. 19, Plates 3 [a] and [b]).

The classification of machines by size is actually a convenient way of sorting out this rather confusing picture. Physical size, though not of major importance itself, is a rough-and-ready indication of the versatility of a digital computer in terms of the number of operations it can perform (addition, subtraction, multiplication etc.). It is also to some extent a guide to the speed and data handling capacity of the machine. The price, which is determined by versatility and capacity, goes in roughly the same proportion —and so does electrical power consumption.

What are generally described as " large " machines cost anything from about £80,000 upwards and take up a complete room for the equipment (see p. 20, Plate 4 [b]). They usually make available fifty or more operations and work in pure binary notation at quite high speeds—the digit pulse rate being 1 Mc/s or above. Often they have a multiplicity of input and output devices and auxiliary storage systems.

At the other end of the scale, the description " small " is applied to machines of about the size of a desk or grand piano (see p. 176, Plate 24 [a]) or even smaller. They are in the £20,000-and-below price range, and offer up to about ten operations, usually working in straight decimal or binary-coded decimal. The digit pulse rate tends to be low—in the 50–100 kc/s region. Machines with built-in programmes determined by the wiring connections (sometimes called " calculators ") fit conveniently into this category.

Between these two extremes comes the " medium-sized " computer (see p. 176, Plate 24 [b]), a type which is now proving very popular because of its moderate price (in the £20,000 to £80,000 region) and adequate number of operations (usually about 20 to 40) for most practical purposes. The number representation is often binary-coded decimal, while the digit pulse rate might be anything from 50 kc/s to 2 Mc/s. Medium-sized computers are manufactured in greater quantities than any other machines (almost on a mass-production basis), and as an example the number of installations of one particular model (the IBM 650) is already approaching the 1,500 mark.

Referring back to the block diagram, then, the arithmetic unit and the control unit between them probably account for the bulk of the electronic circuitry. In a large machine this circuitry will be housed in a series of cubicles—perhaps five or six—which might be arranged separately or fixed together. In a medium-sized machine there will be no obvious division into cubicles but just a single large cabinet with doors or detachable panels. The small computer has much the same arrangement, and the single cabinet usually carries the input and output equipment as well.

Methods of mounting the electronic circuitry vary considerably in detail, but are generally based on detachable chassis which often form definite functional units in the operation of the computer. These can be removed easily for testing or maintenance, and when they are put back in the machine all the electrical connections are made automatically by plugs and sockets on the chassis frame and main frame.

A system of construction which is becoming very popular is based on small, flat plug-in units or circuit cards arranged in racks rather like books in bookshelves (see p. 17, Plate 1 [b]; p. 30, Plate 6 [a]; p. 174, Plates 22 [a] and [b]; and p. 175, Plate 23). With this system, each unit might hold two or three valves or perhaps a dozen smaller computing elements (transistors, diodes, magnetic-core devices) and in a complete machine there could be a hundred or more of such " packages ", as they are called.

The main purpose of all such standardized plug-in systems is, of course, to achieve maximum continuity in the operation of the computer. If a particular part of the circuit develops a fault there is no need to shut down the machine in order to put it right. The plug-in unit containing that part of the circuit is simply pulled out and replaced by an identical spare one. The faulty unit can then be repaired while the computer is still in operation. In addition, if each " package " is designed as a unit with a particular logical function, it is possible to build up different computing assemblies from these basic components, rather on the " Meccano " principle. This, of course, makes the computer a very flexible and versatile machine.

Reverting to the " control unit " in Fig. 8.7, the actual form which this section takes depends very much on the size of the computer. There are, for example, the small " calculators " in which the sequence of operations is fixed by the circuits and connections. These are usually business machines designed for performing a few simple operations like addition, subtraction and multiplication. Although the circuits cannot be switched while computing is in progress, they can be altered by hand before the machine starts working; in other words, the computer can be set up in advance for whatever type of operation is required. This is done by a small plug-and-socket panel corresponding to the " patch panel " of an analogue computer (see Chapter 5).

Between the fixed-programme computer and the more versatile stored-programme machine is an intermediate type in which the programme is stored, not in the form of code digits fed in with the other numerical data but as patterns of connections which are " scanned " in turn by the computer. The required patterns are formed manually by an operator inserting plugs into a plug-board which make connections underneath between groups of wires.

Fig. 10.1 shows one type of layout for such a plug-board. Each of the 15 horizontal row of sockets represents one instruction. A single-address machine is assumed here (see p. 131) and the instruction is divided into an operation part, consisting of one code symbol, and an address part consisting of two decimal digits (tens and units). Thus the instruction in the first row, C17, might signify: " Clear the accumulator and transfer into it the number stored in register 17 ". Similarly, A28 in the second row might mean: " Add to the contents of the accumulator the number in register 28 ". And so on with S19, which might be a " subtract " operation, and T23, which might be a " transfer " operation.

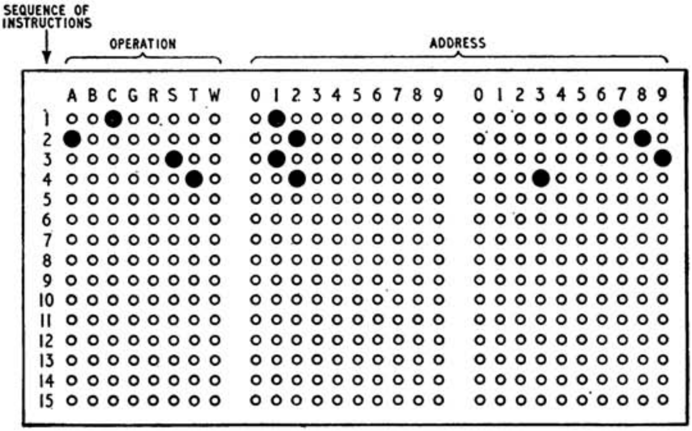


Fig. 10.1. One method of storing a programme of instructions by plugs inserted into a plug-board

Each pattern of connections formed underneath each row on the board is scanned in turn by the computer, starting with No. 1 and passing through the complete sequence of instructions. In fact the sets of connections are the actual electrical paths along which the digit pulses travel between the computing elements, and each set becomes effective as an instruction when it is " applied " to the computing elements. The process of " applying " the connections in sequence is usually controlled by an electronic counter or similar device which performs much the same function as the control register (see Fig. 8.5) in the single-address machine described in Chapter 8.

The particular advantage of this plug-board system is that it provides a simple and inexpensive form of programming for small and medium-sized computers whose applications do not call for long sequences of instructions. It is used, for example, in the small desk-type machine in Plate 24 (a), p. 176. As well as being simple, it is directly under the eye of the operator, who can follow what is happening by the disposition of plugs and even make changes in the programme during the computation.

If a particular programme is likely to be required over and over again with different sets of numerical data, a record of the plug positions can be kept as a pattern of holes in a card, through which the plugs can be inserted into the board. Some computers take the idea one step further by using a conventional punched card for the purpose and making the required connections automatically through the punched holes. This cuts out the labour of inserting plugs by hand and in fact the punched card is " read " in conventional fashion at a special programme input separate from the numerical data input of the machine.

Very little of the electronic circuitry in the arithmetic and control sections of the computer can actually be identified with the work it performs—except perhaps the coders or decoders in the control section, which might reveal themselves as grids of wires (see p. 173, Plate 21 [b]) corresponding to the grid of connections in the circuit diagram (Fig. 8.6). When one comes to the storage unit in the block diagram, however, it is far easier to recognize what the equipment is and what it does. In a large machine it is likely that the backing-up store will occupy one whole cubicle to itself. It may take the form of a magnetic drum (or several drums) surrounded by reading and writing heads and driven by an electric motor, the rest of the cubicle being filled up with amplifiers for the heads and circuits for controlling the rotational speed.

The mechanical features of two typical magnetic drums can be seen in Plates 17 (a) and (b), p. 153. The narrow drum in Plate 17 (a) rotates on a horizontal shaft at 4,600 r.p.m. and stores 94,208 binary digits, while the longer one with the vertical shaft in Plate 17 (b) rotates at 3,750 r.p.m. and stores 665,600 binary digits. It will be noticed that the base-plates and mountings are made very heavy and rugged. This is done to minimize the effect of any external shock or vibration which might temporarily upset the balance of the drum and throw it against the read/write heads, for these have an extremely close spacing from the recording surface (only 0·0005 in. in Plate 17 [b]). For the same reason the bearings have to be very high-grade types, while the drum itself is machined and balanced with great accuracy.

In Plate 17 (b) the drum proper is formed by a copper tube, 12½ in. high and 10 in. in diameter, with a nickel recording surface. The tube has two end-plates incorporating ball-races and the whole structure rotates on a fixed vertical shaft which is held at either end by the outer casing. This shaft is actually hollow, and around it are some windings for the electric motor system which drives the drum. The other windings are attached to the end-plates of the cylinder. Additional windings for braking the drum are incorporated.

The read/write heads are mounted in slotted strips on the outer casing and are pressed in towards the drum by springs, while their exact spacing from the recording surface is adjusted by nuts on threaded rods. Each head consists of a Radiometal core or armature having a read coil (two turns) and a record coil (one turn) and the appropriate two gaps by which the magnetic flux is impressed on the drum. The heads are spaced on the mounting strips at intervals of ¼ in., but by using eight such strips around the periphery of the drum, each displaced vertically from the previous one, the recording tracks are made only $\frac{1}{32}$ in. apart and the whole of the surface is utilized. In another magnetic drum the same effect is achieved with quite a small number of heads by moving them up and down electromagnetically as required.

From the previous chapters it will be clear that the rotation of the magnetic drum has to be maintained at a constant speed so that the pulses recorded on it will be in exact synchronism with the pulses from the clock-pulse generator. This difficulty could be avoided by making the drum itself the generator of the pulses. It is more common, however, to control the speed of the driving motor by a servo system which continuously measures the relative phases of the clock pulses and the drum pulses and applies appropriate correcting signals to reduce the phase error to zero.

Another common type of backing up store is the magnetic-tape recording machine, and this is also frequently used for input and output purposes. A magnetic-tape store or input/output equipment resembles, as one might expect, a magnetic-tape sound-recording machine. Usually the tape is stored on two reels, and on being wound from one to the other it passes across a reading and/or writing head. The tape-driving mechanism, however, is generally rather more complicated as it must include facilities for high-speed starting, stopping and reversing. In one machine, for example, the length of tape between the reels passes over two driving capstans which rotate continuously in opposite directions. The tape is pressed into contact with either one or the other by rollers operated by solenoids, thereby driving it in either one direction or the other. The action of starting, stopping or reversing takes only 10 milliseconds.

Comparable high speed operation is given by another system in which the tape is " sucked " on to either of the driving capstans by a vacuum " applied " through ports in the capstans. More precisely, atmospheric pressure causes the tape to be pressed against the selected capstans and so be driven in the required direction. The selection of one capstan or the other is done by an electro-pneumatic switching system.

In all these high-speed tape drive systems it is important that the tape should offer an absolute minimum of inertia when it has to be started up quickly by one of the capstans. This means that the tape cannot pass directly on to the heavy spool but must have a free intermediate length which will move easily when the drive operates suddenly. Usually this requirement is satisfied by two loops of tape which hang down in boxes or channels adjacent to the two capstans. It is important, of course, that the drive of the take-up spools should be continuously adjusted so that the two loops are always maintained in existence. This is done by a servo system controlled by an arrangement of photocells and light beams in the loop boxes. The lengthening and shortening of the loops causes the photocells to be masked and unmasked, and the resultant signals are used to control the spool drives so that the loops are maintained within certain limits.

A typical magnetic-tape storage unit based on the above principles is shown in Plate 19 (a), p. 171. The two $\frac{1}{2}$-in. tapes run at 100 in./sec. and have a capacity of 23 million binary digits each.

Small or medium-sized digital computers do not usually have a backing-up store as standard equipment (although it depends on the application). They are frequently designed around a magnetic drum, which not only serves as a main store (corresponding to the 32 delay lines in Fig. 8.5) but provides individual tracks to act as separate registers and accumulators. Here the principle is a regenerative one, as in the acoustic delay-line store. At each revolution of the drum the digit pulses from a reading head are amplified, reshaped and fed back to the writing head on the same track.

The only difference is that the storage medium itself is circulating instead of the pulses through the line, and the delay is provided by the time required for the drum to revolve from writing head to reading head.

Where, however, a magnetic drum (or drums) is used as a backing-up store, the computer usually has a somewhat higher speed system for a main store. It might well be a magnetic core matrix, a bank of acoustic delay lines, either nickel or mercury, or a group of cathode-ray tube stores. Some of these devices have already been illustrated and would be immediately recognizable in the equipment of a computer. Mercury delay lines are normally in steel tubes or containers with quartz crystal transducers, and might be housed in a thermally insulated chamber to reduce the effect of temperature changes on the delay.

As mentioned in Chapter 8, the two most common input and output systems are based on punched cards and punched paper tape. The punched cards are the conventional kind used in well-established business machines. The fact that they are separate units, as distinct from continuous rolls of paper or magnetic tape, offers several practical advantages. In the first place, each card can represent a separate entity, like a man on a payroll, a standard period of time such as a day or a week, a particular object undergoing a manufacturing process. This makes for great flexibility in handling by human operators or automatic machinery. Secondly, the cards can be easily read by human operators whenever it may be necessary. Thirdly, they can be passed through existing office machinery for performing operations like sorting, checking, selecting, merging and printing-out the data on to paper documents. It is not surprising, therefore, that many business computers have been designed specifically for punched-card input and output.

A typical punched card, relating to a worker on a payroll, is shown in Fig. 10.2. Ignoring for the moment the annotations, there are ten horizontal rows and 80 vertical columns in which it is possible to punch holes. The general principle of number representation is that the vertical columns stand for decimal places, with units, tens, hundreds and so on working from right to left as is the normal convention on paper. At each decimal place the particular digit to be represented is selected from the ten printed numerals, 0 to 9, in the column and a hole is punched there.

Theoretically this makes it possible to record on each card one number with 80 decimal places. In practice such a thing is hardly ever called for. It is more useful to be able to represent several numbers on a card, so the card is divided into groups of columns, or *fields*, by vertical lines, as can be seen from Fig. 10.2. Each field can then contain a separate number, and a whole mass of numerical data (in this case relating to the worker's pay) can be punched on one card. As an illustration the number 5,248 is punched into the field labelled " Check No."

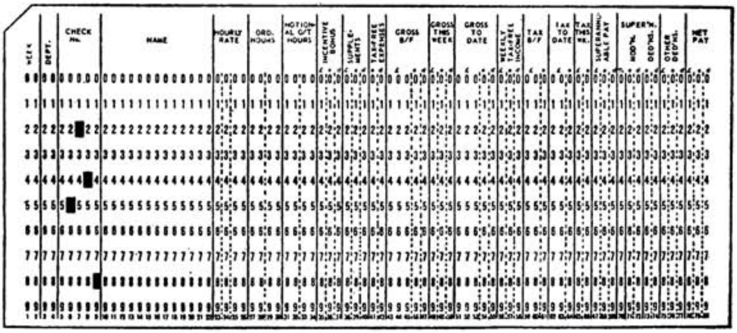


Fig. 10.2. Layout of a typical punched card with 80 columns. The vertical lines divide it into fields, and the number 5,248 is punched into one of them

Incidentally, where pounds, shillings and pence are recorded, the 10d. and 11d. required in the " pence " column are usually punched in two extra horizontal rows (not marked) above the " 0 " row. Letters of the alphabet, required for names and so on, are recorded by means of a numerical code. Usually, each letter is represented by a particular pair of numerals punched into the same vertical column.

The cards are punched by hand, column by column, by a keyboard machine of the type shown in Plate 25 (a), p. 177. They are fed into the computer by means of a card reading equipment (see p. 177, Plate 25 [c]) which may well include also a page printer for recording the computer's output. Modified tabulating machines have sometimes been used for this purpose. In the card reader the general principle of operation is that the card is fed in by rollers (usually the top or long edge first) and meets a row of electrical brushes, one for each vertical column. These brushes " sense " the card row by row, and wherever a hole is punched they pass through it to make contact with a metal roller underneath. This produces electrical signals representing the decimal digits, which are stored in readiness to be converted into binary form for the computer. The cards are read at a speed of about 100 per minute.

Converting the punched-card type of number representation into the computer's binary notation is quite a complex business, and various methods are used, depending on the type of computer. Where the machine works in pure binary notation it is possible to employ a subroutine in which each decimal digit determines the number of successive additions (in the accumulator) of the binary equivalent of the decimal place. For example, if a hole is punched in the " hundreds " column of the card at position " 3 " (representing 300), the binary equivalent of one hundred, 1100100, is added into the accumulator three times to give 100101100 (or 300). This method is explained more fully in Chapter 11.

If the computer works in binary-coded decimal, it is possible to use a system in which clock pulses at timing intervals representing decimal places (see Fig. 7.9) are gated or selected by the card sensing contacts. Here, if a hole is punched in the " hundreds " column of the card at position " 3 ", it gates a " hundreds " clock pulse and feeds it to a wire corresponding to the " 3 " row on the card. This " hundreds " pulse on the " 3 " wire is then coded into binary form and emerges from the coder as a simultaneous pattern of pulses and spaces on four wires (actually 0011) still in the " hundreds " timing interval.

Where punched cards provide the input medium of a computer they are often used for the output as well—either fresh blank ones or the original input cards, depending on the type of application. In some machines the output mechanism is an electro-mechanical card punch. An alternative device is the line printer (mentioned earlier) as used in tabulating machines. This prints a complete line of type at a time (letters or figures) on a roll of paper, which is moved forward after each line is printed. Where the input cards have 80 columns, the line printer has 80 corresponding positions in its line of type, and the required characters to go in these positions are selected by electro-magnets fed from the computer output. Some of the latest line printers will operate at speeds as high as 600 to 1,000 lines per minute.

A particularly interesting type of page printer is shown in Plate 21 (a), p. 173. This has abandoned the conventional method of printing from type founts and uses reciprocating styli to build up the letters and figures from patterns of inked dots. As a result it can print on a page at the rate of 300 lines per minute with 140 characters in each line. A storage system holds the necessary information for constructing 50 different characters and these are called up automatically by code signals fed in from punched cards.

In practice, the full 80 columns of the punched card are not always utilized for input and output purposes, because of limitations set by the capacity of the computer's storage registers. Small or medium-sized machines often use something in the region of 30 to 50 columns. The conventional method of number representation shown in Fig. 10.2

is almost always used for commercial computations, but sometimes different systems are found more convenient for scientific work.

Just as punched cards were in existence before computers came along so was punched paper tape. This medium is, in fact, standard teleprinter tape, the characters being represented by different combinations of five (or sometimes seven) holes punched in rows across the tape. It so happens that combinations of five holes can also be used to represent numbers in binary form. The section of punched tape in Fig. 10.3 gives an example of this, from which it can be seen that the five information channels represent significant places in binary notation (2^0, 2^1, 2^2, etc.) just as the columns of the punched card represent significant places (10^0, 10^1, 10^2, etc.) in decimal notation. The continuous line of smaller holes does not convey information but is provided for drive purposes. In this particular example the numbers are in binary-coded decimal (each decimal digit being coded into binary in a separate row of holes) while the letters are actually in the International Telegraph Code. Most computer manufacturers, however, have their own particular codes which are designed to suit their machines and may only take the general form of binary-coded decimal (one character per row of holes).

One great advantage of punched paper tape as an input/output medium is the continuous nature of the record. This avoids any possibility of the numerical data getting out of proper order. It is also a cheap medium, giving a large storage capacity in quite a small volume, and is very convenient for editing purposes. Whether it is used in preference to punched cards, however, depends very much on the computing application.

The input mechanism for punched tape is usually a high-speed tape reader (see p. 178, Plate 26 [a]) while the output device is a tape punch. Sometimes a modified teleprinter or electric typewriter is used for printing the output data directly on a page, but the tape punch has the advantage of being faster in operation and does not limit the speed of the computer so much. A modern high-speed tape punch working at 300 characters per second is shown in Plate 27 (b) (p. 203).

The tape reader in Plate 26 (a), p. 178, is actually a photoelectric device which will operate at speeds up to 200 rows per second. Light from an electric lamp is projected through the holes in the tape to form images of the lamp filaments on a group of small photocells. Each cell is connected to a single-valve amplifier which gives an output signal of zero volts when a hole is read and more than 20 volts (negative) when there is no hole.

For feeding the tape through the reader a friction-drive system is used. In this the driving roller is mounted on one of the output shafts of a differential gear which has an electric motor driving its cage. Also on this output shaft is an electromagnetic brake which stops the drive when a signal is applied to it. On the other output shaft is another electromagnetic brake, but when this is applied it has the effect of transmitting the drive from the motor and cage through to the roller output shaft. Thus the friction-drive roller can be started or stopped depending on which of the two brakes is energized, and this is actually the means by which the tape is moved past the reading photocells one row of holes at a time. To ensure that the tape always stops with the holes in the correct reading position, the extra line of small holes in the centre is used as a reference. Light passing through these small holes actuates a photocell which produces signals for controlling the " stop " brake.

In a simpler design of tape reader, operating at speeds up to 20 characters per second, the tape is driven forward by means of a sprocket wheel whose teeth engage with the line of small holes. This wheel is moved round in steps by an electromagnetically-operated pawl and ratchet system. Each pulse applied to the electromagnet causes the tape to

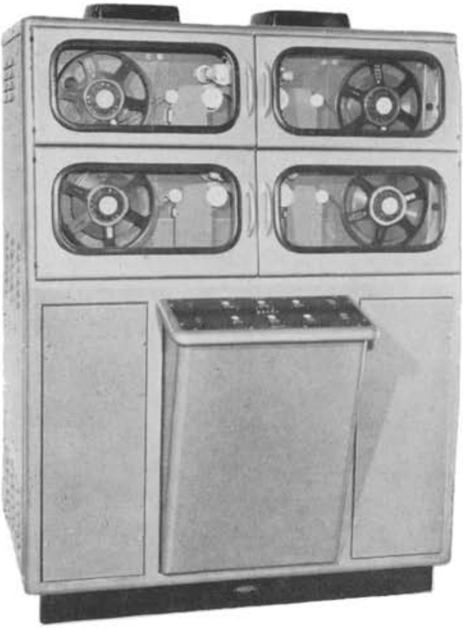


Plate 19 (a). Magnetic-tape storage equipment using two tapes running at 100 in./sec (Decca)

Plate 19 (b). Special typewriter for recording input data directly on magnetic tape

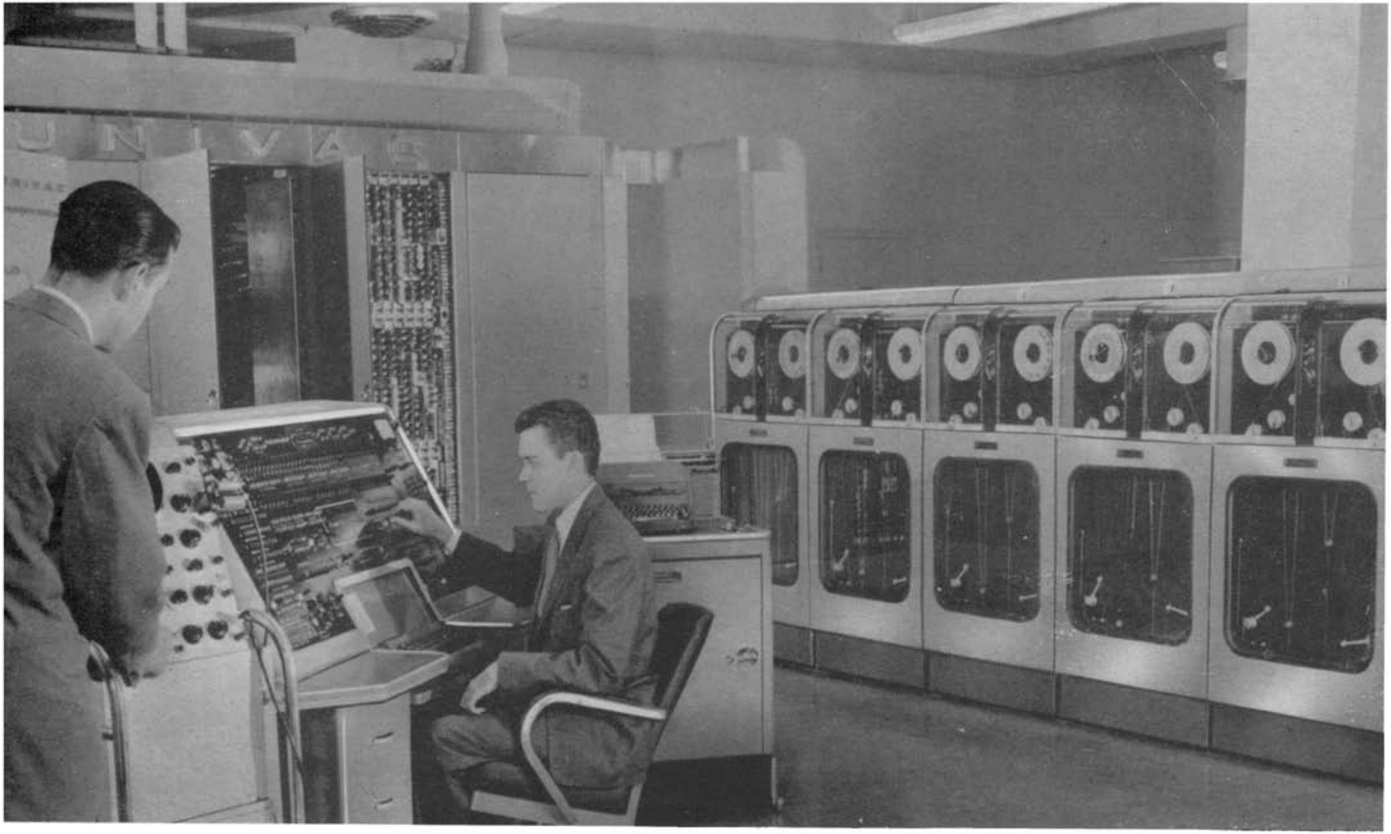


Plate 20. *UNIVAC digital computer in the U.S.A., showing a bank of magnetic-tape storage units on the right*

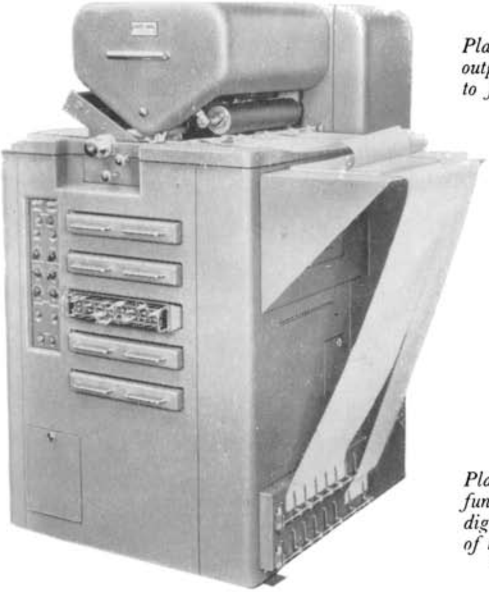
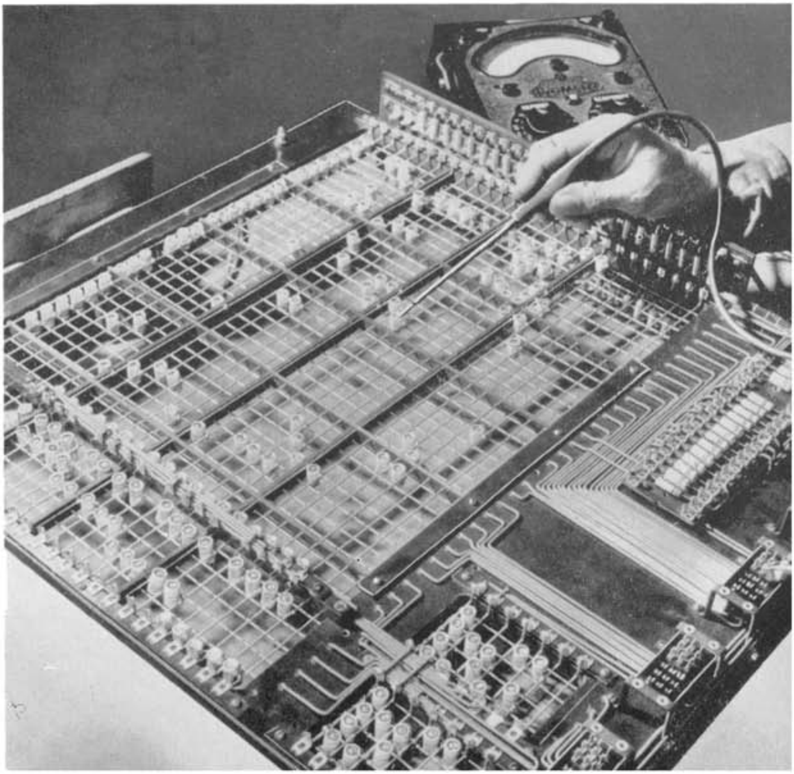


Plate 21 (a). Samastronic high-speed output printer using oscillating styli to form characters as patterns of dots

Plate 21 (b)—below. Testing the function decoder, or "tree", of a digital computer. The physical layout of the device has a close resemblance to the circuit diagram of Fig. 8.7



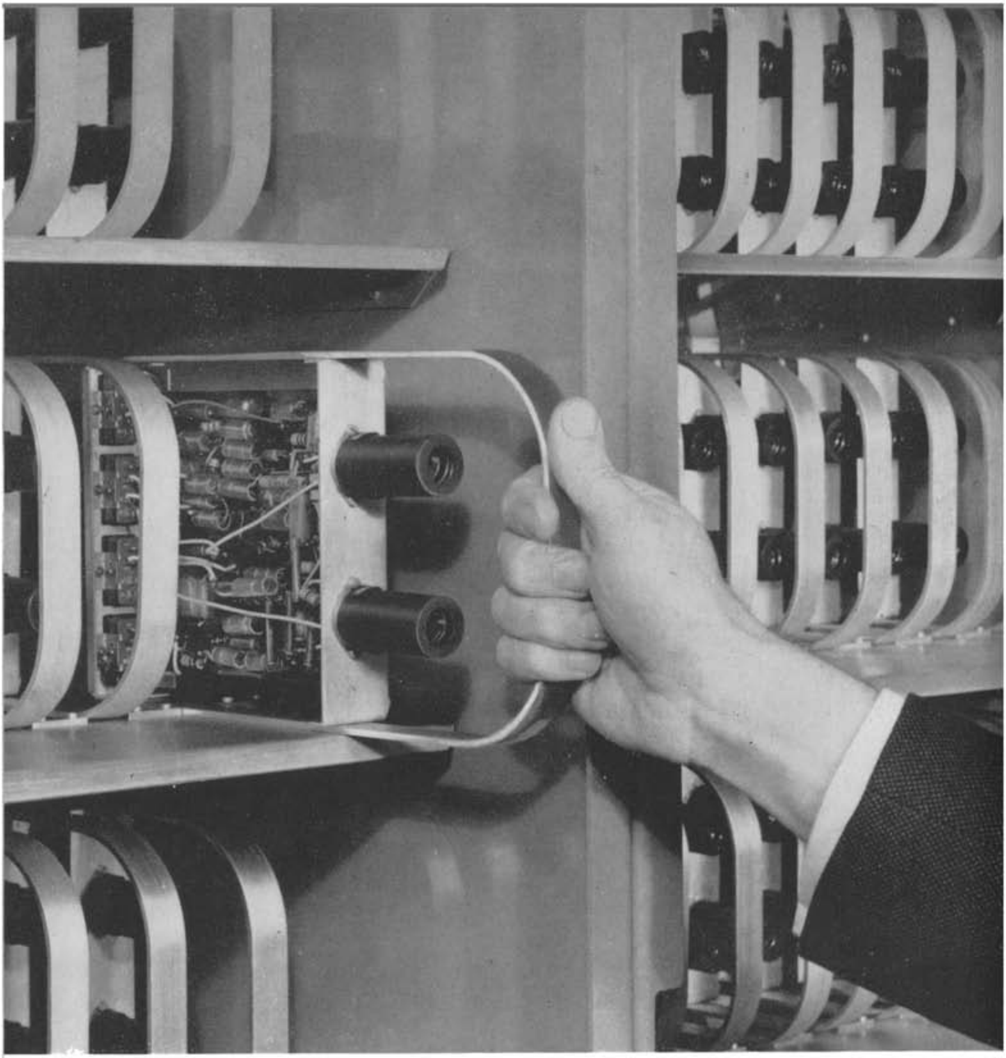


Plate 22 (a) – above. Standardized plug-in units which can be replaced when faulty, as used in a " packaged " digital computer

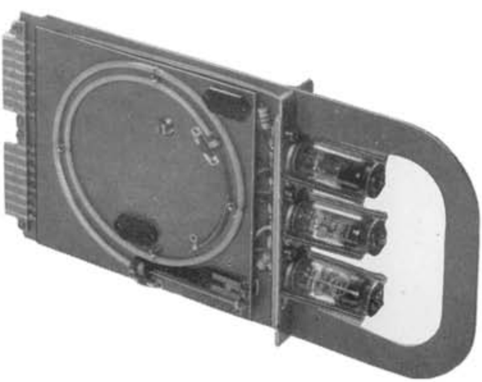


Plate 22 (b). A single unit of the above type from the machine in Plate 1 (b). It is actually a delay-line store and the magnetostriction line can be seen arranged in a wide coil

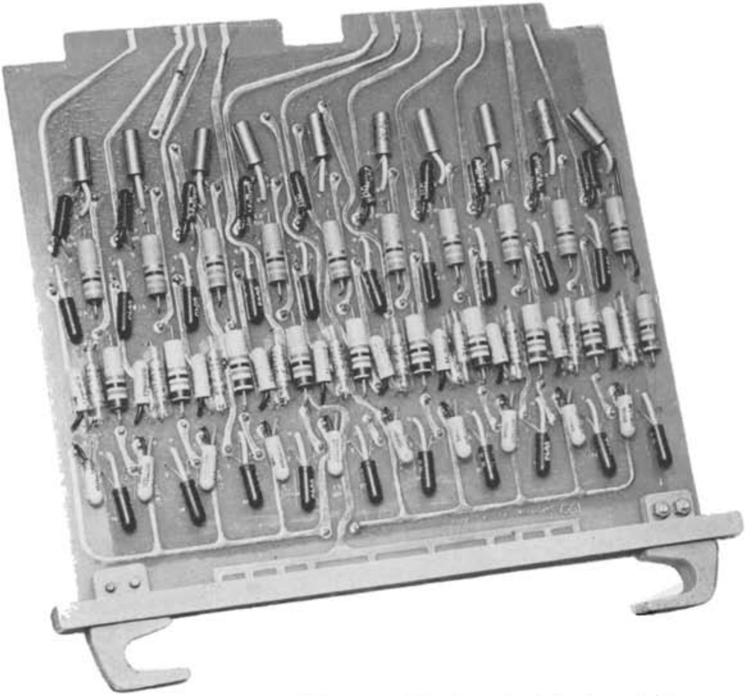
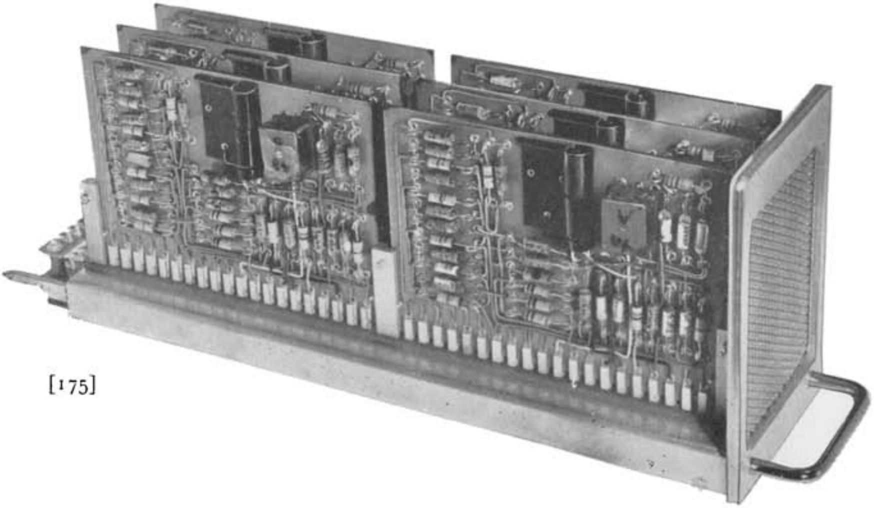


Plate 23. Plug-in printed-circuit card for an E.M.I. Electronics magnetic-core/transistor computer (top), and circuit cards mounted on drawer-type chassis (below) which plugs in at the rear



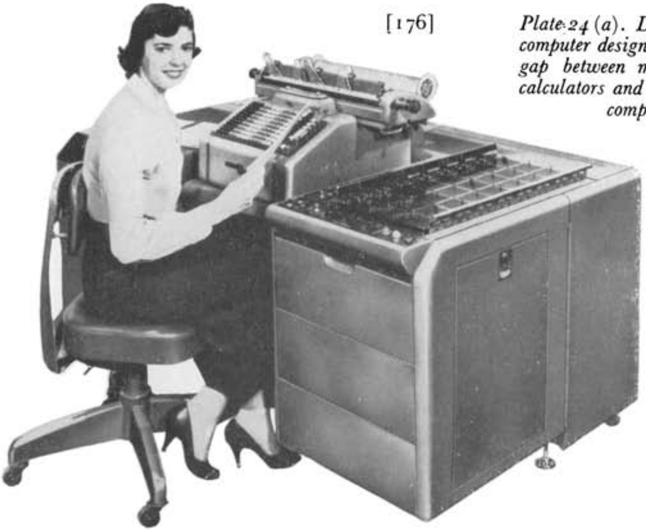


Plate 24 (a). Desk-size digital computer designed to bridge the gap between mechanical desk calculators and large electronic computers

Plate 24 (b). DEUCE digital computer, based on the design of the Pilot ACE, at the English Electric Company's London computer centre

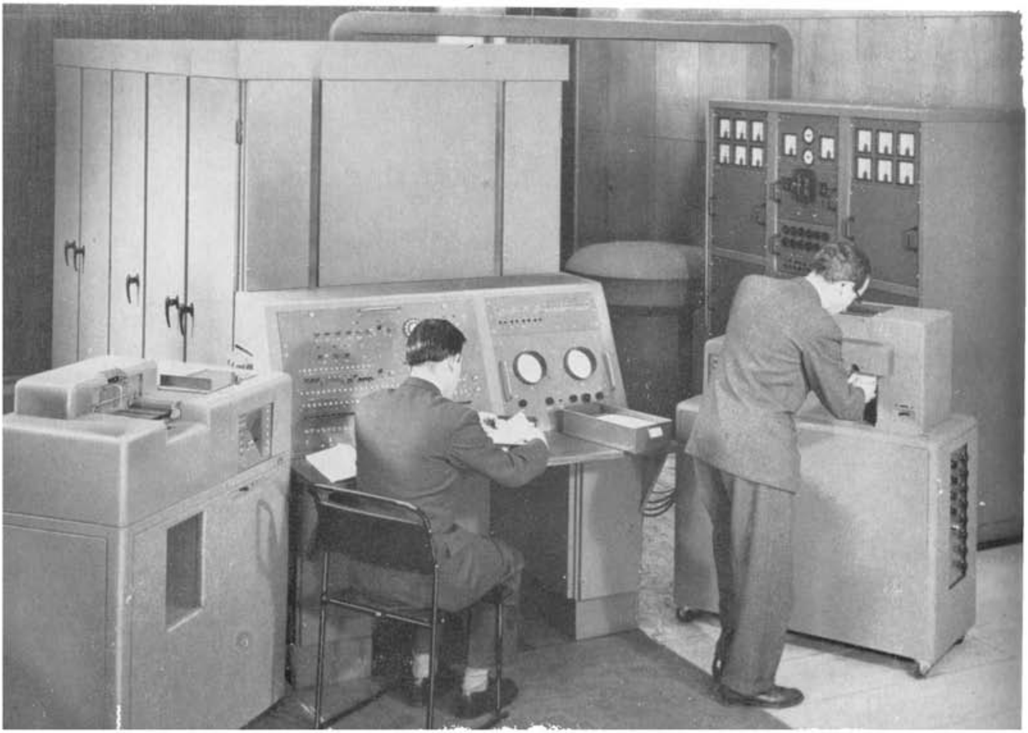


Plate 25 (*a*)—*above. Machine for making the perforations in punched cards by which information is fed into some types of digital computer*

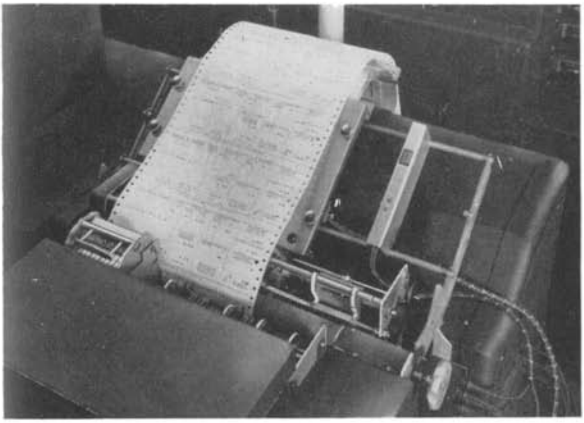
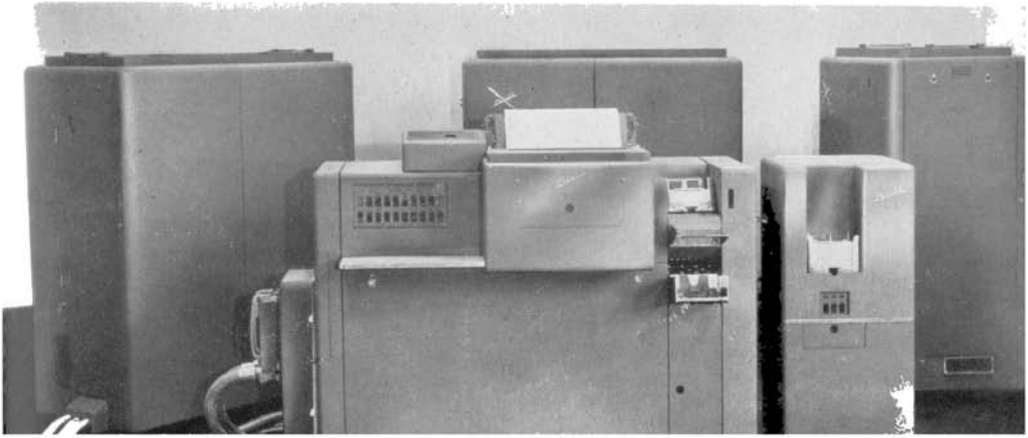
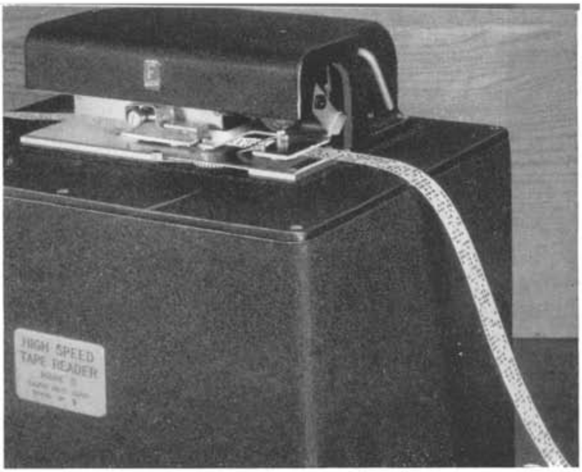


Plate 25 (b)—left. Device for printing output information from a digital computer, one complete line of type at a time

Plate 25 (c)—below. Small French digital computer produced by the Compagnie des Machines Bull





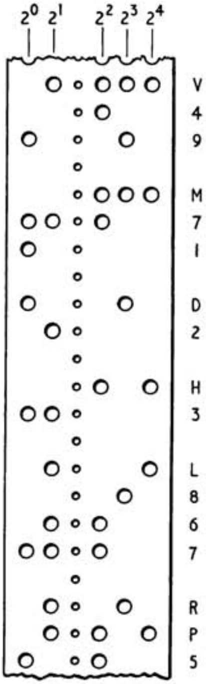
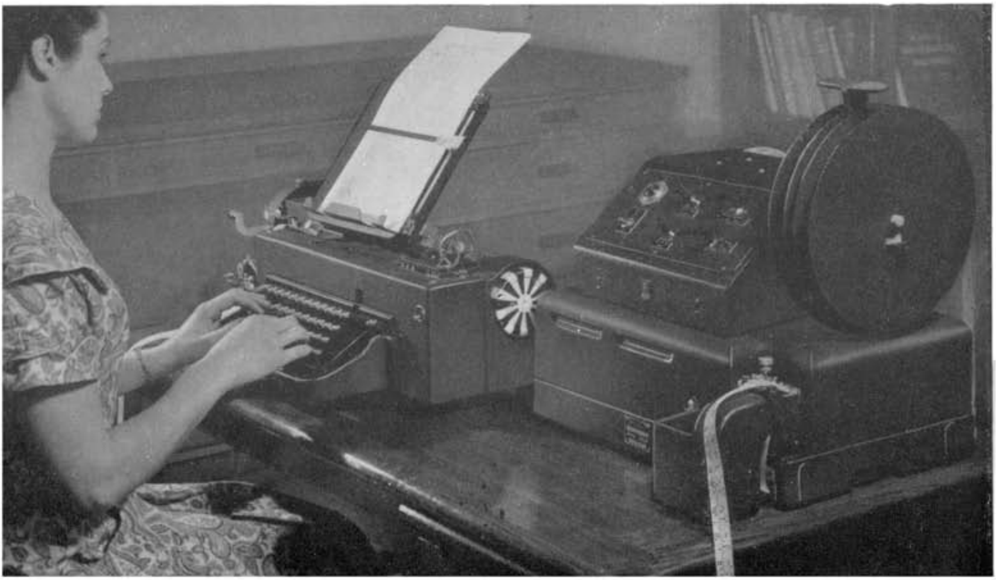


Plate 26 (a)—above. High-speed tape-reading mechanism for digital computers. Fig. 10.3, to the left, is a section of punched tape, showing how letters and numerals are represented by different combinations of holes and spaces in rows

Plate 26 (b)—below. Equipment for perforating and editing punched tape. The machine on the left is a modified teleprinter



move forward one row of holes. The recorded information is read by five spring-loaded sensing fingers which pass through the tape whenever holes are presented to them. The resulting movements cause contacts to be closed which generate output voltage pulses.

For putting the information on the tape in the first place, before it goes through the reader, an equipment of the kind shown in Plate 26 (b), p. 178, is commonly used. This consists of a modified teleprinter whose output signals, representing digits by a five-unit code, are fed into an electromagnetic tape-punching device. It does the equivalent job, of course, to the card punch shown in Plate 25 (a), p. 177.

Another type of input and output system is based on the use of magnetic tape, as already mentioned. In some equipments the input information can be impressed on the tape directly, by means of a special kind of typewriter incorporating a magnetic recording head (see p. 171, Plate 19 [b]), while the output tape directly actuates a line printer. In other computers the magnetic tape system acts as a temporary storage system between the computer and, say, conventional punched-card input and output equipment. The idea of this intermediate system is to enable the computer to continue operating on one batch of data, being fed in and out by magnetic tapes, while the next batch is being transferred from punched cards on to a new input tape and the results from the previous batch are being transcribed from their output tape back on to punched cards. In other words, the machine does not have to stop to wait for the relatively slow input and output operations. Magnetic tape is used for the temporary medium because digital information can be read from it and written on to it at much higher speeds than with punched cards or tape, and it is therefore more suited to the " electronic speed " of the computing circuits.

Various patterns are used for magnetically recording the digits, depending on the width of the tape, the number of parallel recording channels and the design of the computer. In general the tendency is to record the information in blocks of several hundred digits or more. During the read-in process each block is transferred complete to a " buffer " store in the computer, and from there the numbers are selected by the computing circuits in the manner described in Chapter 8. Computing proceeds in the intervals between the transfer operations.

Another " buffer " store is used in a similar way at the output stage. The numerical results of the calculation are collected in it and then transferred in blocks to the output tape. Each block of numbers along the tape is therefore roughly comparable with a block of numbers on a punched card.

A common tape width is ½ in., but the sizes in use range from ¼ in. to 4 in., including magnetically-coated 35 mm film stock. Along one channel of the tape the digit pulses might be recorded with a density of, say, 100 to the inch, and an average reel might contain several million digits in each track. With the tape moving at 200 in./sec the speed of reading or writing could be in the region of 20,000 digits per second.

Recent developments suggest that the digit pulses can be packed as closely as 500 per inch on the tape, which would give more than twice this speed.

This question of the speed of the input and output equipment is important because it is the thing which controls the speed of operation of the whole computer. With any kind of electromechanical devices the " electronic speed " of the computing circuits is to some extent wasted, so there is a great need for developing not only faster equipment of the present kind but entirely new methods of handling the input and output information.

There are, however, qualifications to this, since high speed is not essential to all applications. Where, for example, the computer is used for reducing large masses of data to a few significant results, the output equipment does not have to handle the same quantity of information in a given time as the input equipment and therefore can be quite a slow-speed device. There is no need for high speed, either, in small machines which are only used occasionally and do not have to justify their price by almost continuous operation.

In the small computer which is shown in Plate 24 (a), p. 176, for example, the numbers are fed in by the comparatively slow process of setting them up by hand on a keyboard of the desk-calculator type.

All the functional units in the block diagram have now been discussed, but in looking at a typical digital computer one would see another item of equipment which is not shown on the diagram—the control desk. On this would be found mostly switches, push-buttons, indicator lights, meters and cathode-ray tube screens (see p. 19, Plate 3 [a]; and p. 176, Plate 24 [b]).

The various switches and press-buttons are provided for such purposes as starting and stopping the computer, reading-in programmes, clearing stores and registers and inserting numbers into storage locations by hand. There might be others for use in connection with marginal testing and special test programmes. The cathode-ray tubes are used for observing the contents of the various storage registers, which are selected in turn by rotary switches. In some machines hundreds of small neon tubes are provided for indicating the state of each trigger-circuit. Others have just a single row of lights for indicating the current instruction being carried out.

This chapter can probably best be rounded off by giving some figures which describe the performance of a typical digital computer—as they might appear on a written specification.

The summary on the following page refers to a purely hypothetical medium-to-large-sized computer operating in pure binary, but is comparable with several machines actually on the market.

Speed of operation:	Addition, subtraction and transfer operations take 1 millisecond. Multiplication, 5 milliseconds. Division, 10 milliseconds.
Digit pulse rate:	200 kc/s.
Number length:	40 binary digits.
Main store:	48 single-number delay-line registers. Access time, 200 microseconds.
Backing-up store:	Magnetic drum with capacity of 5,000 numbers. Average access time, 10 milliseconds.
Instruction code:	45 single-address instructions available. Two instructions in one number length.
Input mechanism:	Punched card reader, operating at 100 cards per minute.
Output mechanism:	Card punch, operating at 100 cards per minute. Line printer, operating at 100 lines per minute.
Power consumption:	8 kilowatts.
Size:	Main unit, 8 ft × 2 ft 6 in. × 5 ft. Control desk, 5 ft × 3 ft × 4 ft.

11

PROGRAMMING DIGITAL COMPUTERS

VERY little has been said in previous chapters on the subject of how calculations are prepared for the digital computer. In Chapter 7 it was explained how numbers coming from storage registers could have arithmetic operations performed on them—addition, subtraction, multiplication, and so on. Then in Chapter 8 we saw how a set of stored instructions could be made to switch the circuits of the computer so that it would carry out the various stages of a calculation in the required sequence. So far so good. But there is still a big gap between the initial statement of the problem on paper (e.g., " find the cost of a dozen articles at 19s. each ", as in Chapter 2) and the moment when we find it stored as a pattern of binary digits in the machine, waiting for somebody to press the starting button.

This big gap is bridged by the process known as programming, as already explained. The problem has to be translated into a " language " which the machine can " understand ". Usually the preparation of a programme involves two distinct stages. The first is to break down the written problem into a series of arithmetical or logical operations following one after the other. The second stage is to take these operations and express them in a code which has some significance to the actual electronic design of the computer.

Finally, the encoded programme steps are put into the machine. How this is done will depend on the type of computer one is dealing with. With a small machine one would probably have to set up the programme steps as rows of connections on a plug-board or as rows of holes in punched cards (which would allow connections to be made through them). With a larger computer, the normal procedure is to convert the code into punched holes in cards or paper tape (using a suitable machine) and these are fed as binary numbers into the storage registers of the machine.

If the problem written on paper is already in arithmetical terms the process of programming is straightforward. If it is a mathematical problem, however (like solving a set of simultaneous equations), it has to be converted first of all into a series of arithmetic operations.

Up to this point we have only talked about breaking down the calculation into a series of arithmetic operations and instructions. But of course the calculation also involves numbers, on which the arithmetic operations are carried out, and these also have to be put into the machine. In small computers the numbers are usually set up in decimal form, either on a keyboard or on punched cards, and the electrical versions of them are then used directly. With large-scale machines, however, the numbers are punched into cards or tape as a first stage, perhaps in binary-coded decimal form, and then fed into storage registers inside the computer after being converted into pure binary notation.

The process of reading the numbers from the cards or tape, converting them, and putting them into the storage system is usually done automatically by a sequence of instructions before the commencement of the normal calculating programme. To transfer the programme itself into the machine in the first place, it is usually necessary to have a built-in " initial programme " which reads the coded instructions and puts them into a section of the storage system separate from the " number " section.

To give a better idea of how programming works in practice it is worth while looking at an actual example.

Suppose that in the course of a wages calculation it is necessary to ensure that an employee gets a minimum wage of £8 for the week, even though the hours he has worked at a certain rate of pay may not add up to this. What the calculation entails, then, is first of all to multiply the hours of work by the hourly rate to get the actual earned amount. Then this figure, the product, has to be compared with £8 to see which is greater. If the earned amount is greater, then that figure is taken and used. If, however, the earned amount is smaller, the £8 is taken.

To show clearly the sequence of arithmetic or logical operations required in such a calculation, the usual practice is to construct what is known as a flow diagram. In this case it would be arranged as shown in Fig. 11.1. The sequence is just part of a complete wages calculation, so it is assumed that the numbers for hours, rate and £8 are held in storage registers and that the final answer is also to be transferred to a storage register. The comparison of the hours × rate product with £8 is represented as it would be done in the computer—by subtracting £8 from the

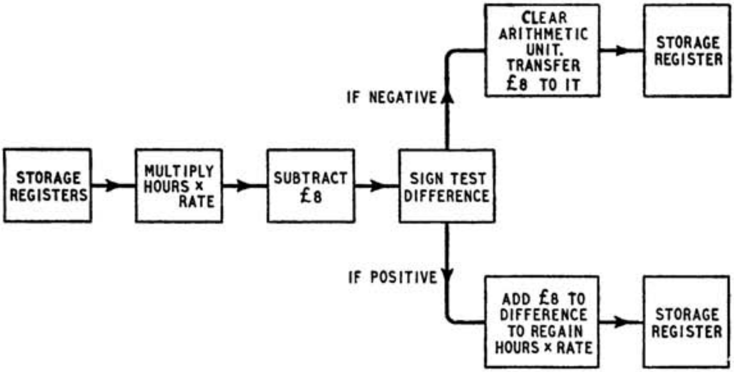


Fig. 11.1. Flow diagram for part of a wages calculation, showing the sequence of the operations to be performed in the computer programme

product and testing to see if the difference is negative or positive. If it is negative the £8 minimum wage is stored and if it is positive the actual hours × rate amount.

Having now assembled the programme steps in logical order on a flow chart, the next thing to consider is how these steps are encoded into " machine language ". The actual form of the code, as already mentioned, depends on the electronic design of the computer. In particular, it depends on whether the operations inside the machine are based on a " single-address ", " two-address " or " three-address " system. The control arrangements described in Chapter 8 use a single-address system, and as this is perhaps the most common, and certainly the easiest to understand, we will assume that it is used throughout this explanation.

The " address " here refers to the location in the main store to or from which the number is transferred. All single-address instructions contain just this location and the operation which is to be performed on the number. No other address is specified, giving the source or destination of the number, because that is already implied by the code. A good many instructions, for example, are concerned with transfer of numbers to and from the accumulator. Here the accumulator is the implied address and it is the *other* address which has to be specified. Moreover, the code often implies as well that in the process of transfer the number shall be added to or subtracted from any number already in the accumulator. Thus a typical instruction might take this form:

> " Transfer the number in storage register 28 to the accumulator and add it to the number already there."

This expresses what is to be done in words, but in order to put the instruction into the computer it has to be translated into code form. One might use the letter " A " to represent the operation, as adding is the principal part of it—but this is only a matter of convenience and any other symbol would serve equally well. Thus the complete instruction in code form would become

A 28

Continuing the idea, a typical list of instructions might look something like this, with the code version on the left and the full description on the right:

C 17	Clear the accumulator of any previous number and transfer into it the number from register 17.
A 28	Add to the number already in the accumulator the number from register 28.
S 19	Subtract the number in register 19 from the number in the accumulator.
T 23	Transfer from the accumulator whatever number is in it to register 23.

These instructions, of course, are only concerned with one type of operation, associated with the accumulator, but in a complete programme many other processes would be required. For example, in a large-scale computer it is necessary to read the numbers from the input cards or tape and transfer them to a storage register. Thus we might have:

R 17	Read the next row of holes from the input card (or tape) and transfer the resulting pattern of binary digits to register 17, at the same time moving the card (or tape) on to the following row.

Here, of course, it is the stored number on the cards or tape which is the implied address.

Again, one might have another set of instructions concerned with the output of numbers from the computer, and if the result were to be printed directly by an automatic typewriter these would have to include instructions for mechanical operations such as letter spacing, line spacing and carriage return. Other instructions might be required for multiplication, division, testing for positive or negative sign, shifting numbers in registers (for multiplying or dividing by powers of 2), transferring numbers from an auxiliary store to the main store, and so on.

The next question to consider is how the list of code symbols is actually put into the machine. With the small "plug-board" type of computers this is a comparatively simple matter. The operation to be performed and the address of the register concerned are selected by inserting plugs into appropriately labelled holes (or perhaps by punching cards with corresponding holes). Each horizontal row of holes represents one instruction, and as an example, the plug-board in Fig. 10.1 of Chapter 10 is set up with the typical list of instructions on p. 187. The inserted plugs make sets of connections underneath the board, and these, when they are scanned in sequence by the computer, cause the appropriate operations to be carried out.

In the large-scale stored-programme type of computer the coded instructions have to be fed into storage registers, as already mentioned, and in order to do this an additional process is required to convert the code into binary notation. A common method of achieving this is to produce a punched tape using a tape-perforating typewriter of the kind described in Chapter 10. When a letter or number key is depressed on the typewriter keyboard, a 5-unit pattern of holes (representing "1s") and spaces (representing "0s") is punched on one row of the tape. The binary code used for this purpose varies from computer to computer, but as an example the letters of the alphabet A, B, C, D etc. could be numbered consecutively 1, 2, 3, 4 etc., and the binary-coded versions of these numbers punched. With numerals, each decimal digit of the number is rendered separately as a row of holes. Thus the instruction C 17 would be converted into the binary-coded form

00011	00001	00111
(C as 3)	(1)	(7)

and would appear on the tape with the coded decimal digits in successive rows of holes. In this way the list of coded instructions compiled from the programme can be put on to the tape and eventually fed into the computer storage system. Corresponding methods are used with punched-card or magnetic-tape input mechanisms.

In the majority of stored-programme computers the coded instructions are put into consecutive registers in the storage system and in the course of the calculation they are taken out in the same ordered sequence. Thus in the hypothetical computer of Chapter 8, with its main store of 32 delay-line registers, a typical list of coded instructions (like that on p. 187, and more) might be stored in a block of 15 registers with serial addresses 01, 02, 03 . . . to 15. This would leave the remaining 17 registers, with addresses from 16 to 32, for storage of the actual numbers used in the calculation.

At this stage, then, we are in a position to see how the flow diagram (Fig. 11.1) of the minimum wages calculation would be translated into a coded list of instructions. To begin with, it will be assumed that the relevant numerical data—hours, rate and minimum wage—are already placed in storage registers with the following addresses (an additional register being reserved for the answer)

hours worked	16
rate in shillings per hour	17
minimum wage figure	18
for answer	19

Working from the flow diagram the list of coded instructions would then be assembled as in the table on p. 190. The instructions would have to be carried out by the computer in the order shown, so the right-hand column gives the sequential addresses of the registers in which they would be held in the main store of the machine.

The first three instructions are quite straightforward, but with J 06 in register 04 something unusual happens. It will be noticed that the normal mode of obeying the instructions

Coded instructions	*Full description*	*Register address*
C 16	Clear the accumulator and transfer into it the number from register 16.	01
M 17	Multiply the number in the accumulator by the number in register 17 and retain the result in the accumulator.	02
S 18	Subtract from the accumulator the number in register 18.	03
J 06	Sign test the difference. Jump to the instruction in register 06 if it is negative. If it is positive continue with the instruction in the next register.	04
A 18	Add to the contents of the accumulator the number in register 18.	05
C 18	Clear the accumulator and transfer into it the number in register 18.	06
T 19	Transfer the number in the accumulator to register 19.	07

in sequence is not adhered to if the number in the accumulator is found to have a particular sign (in this case negative). In fact, the instruction in 05 is ignored and the control sequence "jumps" to the one in 06. Otherwise (with a positive sign), the instructions are carried out in the normal sequence. In short, the programme can be modified according to the partial results obtained in the calculation.

Here, then, is an example of how the "conditional operation" mentioned in Chapters 1, 2 and 8 can be utilized in a practical way. Very often the jump is made to some previous instruction, which has already been obeyed once, and as a result the same cycle or "loop" of instructions can be repeated over and over again. This makes possible repetitive processes, which can be extremely useful in many calculations. The repetition is only brought to an end when a stored number, as a result of the operations performed on it, is made to satisfy some pre-determined condition. A practical example will be given later.

It will be remembered from Chapter 8 that each of the delay-line registers in the main store can only hold a ten-digit binary number, sufficient for an instruction converted into two five-digit binary numbers (such as J 06, ignoring the 0). Instructions like C 16, therefore, which are in binary-coded decimal, could not be accommodated and they would have to be first converted into pure binary. Thus

00011	00001	00110	would become	00011	10000
(C)	(1)	(6)		(C)	(16)

The same would apply to large decimal numbers used in the calculation. In binary-coded decimal, for example, the number 245 would be read into the computer as 00010, 00100, 00101, which is obviously too long to go into a ten-digit store. Converting it into pure binary, however, would give 11110101, which contains eight digits and could, of course, be accommodated.

The conversion is normally done as part of the input procedure. It consists of multiplying each of the binary-coded decimal digits by an appropriate power of ten and then adding the results together. Thus with the number 245, the code version of the 2 would be multiplied by 100 (because it really represents 200), the code of the 4 would be multiplied by 10 (because the 4 really represents 40), while the code of the 5 would be multiplied by 1 (i.e., 10^0) or, in other words, just left. Finally, all three products would be added together. With a punched-card input system the number 245 might be represented by a hole punched in position 2 in the "100s" column, a hole in position 4 in the "10s" column and a hole in position 5 in the "1s" column. A common method of converting this into binary is to take the binary equivalent of 100 and add it into the accumulator two times, take the equivalent of 10 and add it in four times, and take 1 and add it in five times. The grand total then gives the 245 in binary form.

All these operations for decimal/binary conversion are performed in binary arithmetic in the computer, and clearly a whole sequence of instructions is necessary for the purpose. The sequence has to be used over and over again, so it is normally treated as a complete self-contained unit, or sub-routine, on the principle mentioned in Chapter 2. It can then be prepared in advance on punched tape or cards and fed into the store of the computer, to be called on as required. Normally a stored sub-routine is bought into operation by a single coded instruction in the main programme, such as U 09, and afterwards the programme proceeds to the next instruction. Thus, from the programming point of view, sub-routines can be treated as just ordinary instructions—and in fact some programmes are almost entirely built up from them.

Sub-routines are commonly used for multiplication and division, for finding powers and roots, exponentials and logarithms, sines, cosines and other trigonometrical functions, for integration (by summing computed areas of rectangles under a curve)—and in fact for any arithmetical process

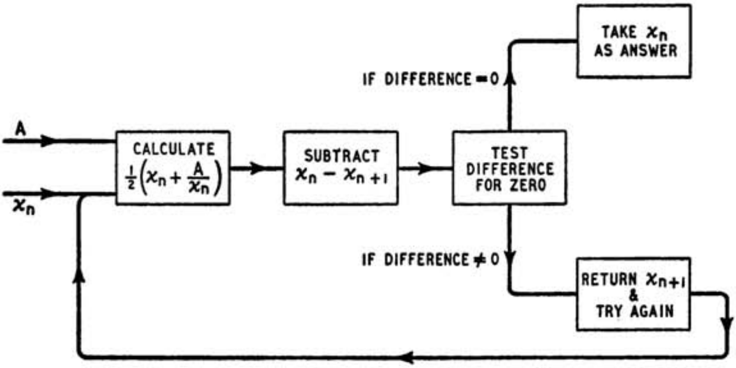


Fig. 11.2. Flow diagram of a sub-routine for finding the square root of a number

which can be reduced to a cut-and-dried set of instructions. As an example, the flow diagram in Fig. 11.2 shows the logical steps required for finding the square root of a number. This works on the principle of calculating successive approximations to the answer, using the result of each calculation in the next one to obtain an even closer approximation. Supposing we wish to find the square root of a number A, the calculation is based on the formula

$$x_{n+1} = \tfrac{1}{2}\left(x_n + \frac{A}{x_n}\right)$$

Here x_n is an approximation to the answer, and by using this in the right-hand side of the equation a value is obtained for x_{n+1}, the next approximation to be tried. This value is put in place of x_n on the right-hand side, a new value for x_{n+1} is calculated, and so on. As the approximations get closer to the answer, the difference between their values (that is between x_n and x_{n+1}) becomes smaller and smaller, until finally, when the answer is reached, the difference is zero—and this is used to stop the process.

In practice, the initial approximation fed in is usually made 1, and although this may be wildly inaccurate as an estimate, the computer does not have to perform many of the cycles of calculation in order to bring it to the right answer. For example, to find the square root of 25, starting with 1, takes only four steps. Moreover, the computer can run through a whole series of such repetitive steps at very great speed and so the laboriousness of the process does not really matter.

It will be noticed that this sub-routine requires a conditional operation of the kind mentioned earlier—only here the future course of action is decided by whether a difference value becomes zero or not zero. Moreover, it will be seen from the flow diagram that the condition of " not zero " demands a " jump " in the sequence of instructions. This time, however, it is not a jump forward, as in the wages calculation, but backwards to an instruction which has already been carried out—thereby forming the " loop " of instructions which is repeated again and again until the " zero " condition is satisfied. This is an example of how the " jump " instruction mentioned earlier can be used to give repetitive operation for a practical purpose.

An important part of compiling a programme for a digital computer is to ensure that the magnitude of every number occurring during the course of the calculation should lie within certain limits imposed by the design of the machine. A number must not be allowed to grow too large or it will " overflow " the register in which it is held and the most significant digits will be lost. Nor must it become too small so that precision is reduced by the loss of the least-significant digits. These matters are quite often attended to by various programming techniques, and by scaling all the data to be used in the calculation and also the results.

An alternative method is to express the numbers in " floating-point " form. This simply means that numbers are stored in two parts: a standardized fractional part, which is multiplied by an " exponent " part with a sufficiently wide range to accommodate all numbers likely to occur in practical computations. As an example in the decimal scale, the number 329,000 might be represented as 0·329, + 6, since it is the product of 0·329 and 10^6. In the actual computer these two parts of the number are each represented by a group of binary digits and are stored separately. This arrangement ensures that all numbers are automatically held and handled within the machine in such a way that they cannot overflow the capacity of their registers, nor be lost through becoming too small. The range of numbers which can be accommodated is, in practice, unlimited. In scientific calculations particularly this " floating-point " method can be extremely useful.

One of the latest developments in the sphere of programming is what is known as " automatic programming " or sometimes " automatic coding ". The idea of this is to simplify the whole procedure so that any person needing to use a digital computer for a calculation can operate the machine without having to call on the services of a specialist programmer. In other words, the purpose is to eliminate the tedious business of translating instructions written in plain English (as on the flow diagram, Fig. 11.1) into the special coded form which constitutes the " machine language ". The actual translation into " machine language " is, in fact, done by the computer itself by means of a " translation programme ".

This whole idea arose from the fact that digital computers can be used to translate written matter from one language to another (see Chapter 12). In practice it means that the computer has to " recognize " the patterns of letters in certain English words (and also symbols like +, −, ×, ÷) and produce, in response to them, binary code symbols which can be used to control the operation of the computer.

Thus, instead of having to use a programmer's code, like the C16, M17, S18, etc., in the table on p. 190, it becomes possible to feed into the computer simple statements written in English. A typical example might be WAGE = TIME × HOURS, as could be taken from the flow diagram (Fig. 11.1). In preparing a programme for the computer, these words, and all other instructions required, are tapped out on the keyboard of, say, a paper tape punching equipment (see Plate 27 [b], p. 203). This converts them into the binary code form of patterns of holes and spaces on punched paper tape, which is fed into the computer. The information, however, is still essentially a representation of written words, and so the translation programme (already stored in the computer) has to be brought into play to convert it into a normal " machine language " programme which will operate the electronic switching in the computer's control system.

Of course, even the English words used in the automatic code have to be selected by the human operator according to certain rules, since the translation programme in the computer is only capable of recognizing words and special groupings of words (phrases) which appear in its stored " dictionary ". It recognizes when a word ends by the space which follows and when a phrase ends by the punctuation mark.

The translation programme does not, however, make use of all the written words which are fed into the computer. It only recognizes the key, or operative, words. All others are redundant, but are kept in by the human operator when preparing the programme for the sake of readability. For example, one part of Fig. 11.1 after the " sign-test difference" block might be written by the programmer: " IF X IS POSITIVE GO TO B ". Here the operative words are " IF ", because it signifies that a decision is to be made, " POSITIVE ", because it is a condition on which the decision depends, and " B ", because it names the next appropriate part of the programme. The words " IS ", " GO " and " TO " are redundant in this context and are not recognized. The " X " is a numerical value, provided by the computer, from previous calculation, and does not have to be translated.

Generally speaking, automatic programming requires greater storage capacity in the computer and more time for operation than conventional programming techniques. These are the penalties to be paid for the gain in simplicity. Now that storage capacities of machines are becoming larger it is likely that the principle will be applied on a much wider scale than it has been so far.

12

APPLICATIONS OF DIGITAL COMPUTERS

At one time it was quite sufficient to divide the applications of digital computers into " business " and " scientific " categories. To a large extent this simple classification is still valid, but in the last two or three years automatic computing methods have proliferated into so many subjects and spheres of activity that it now gives a very inadequate picture. There are some applications which do not fall conveniently under either of these headings.

Many people are confused by the term " data processing " which has now come into common use. It has been indiscriminately applied to anything to do with computing, so that in some quarters the two terms have become almost synonymous. Certainly it cannot be denied that data are processed inside a computer. But in fact the total field of activity known as " data processing " contains many other techniques and operations besides computing. The true state of affairs (at the time of writing, at any rate) is that digital computing can be used for, and often forms a large part of, data processing work. In practice this means that the computer is required more for its ability to handle large masses of data at high speed than for its ability to do complex calculations.

Data processing, then, is one of the main groups of applications of digital computers. It is characterized by the fact that the computer is used continuously, or at very regular intervals, for the same sort of work—like preparing a factory payroll once a week. The computer is often designed specially for the data processing type of work, although sometimes general-purpose machines are employed.

Another group of applications which can be considered as a whole are specialized calculations. These calculations might be connected with any field of activity—business or industry, science, technology or military operations—but they are distinguished by the fact that, as distinct from data processing, they use the computer in an intermittent way. A general-purpose machine in a computing centre might well be used for computing sales statistics in the morning, working out a weather forecast in the afternoon and designing a turbine in the evening.

The newest group of applications is where the computer becomes just a component of a larger automatic system—it might be a process control system in a factory or an automatic navigation system in an aircraft. Here the input to the computer is not provided by punched cards or tape but by electrical waveforms which come directly from components of the control system. Similarly, the results of the calculations are not recorded for human beings to take action, but the digit pulse signals representing the results are used directly (or after transformation) as actuating signals in the control system.

Let us now look at a few typical applications. First, in the sphere of data processing. Undoubtedly the biggest and most important role of the digital computer here is in the mechanization of office work. There are between two and three million clerical workers in Great Britain at the moment and the number is increasing at a fast rate. A single large office organization, like a Government ministry or an insurance office, might contain anything between 10,000 and 20,000 clerks. And the main work of all these people consists mostly of sorting, filing, summarizing, calculating, abstracting, comparing and writing information on forms, bills, receipts, orders, cheques and so on, using purely mechanical routines which demand the very minimum of constructive thought.

It is true that calculating machines have been used for years in offices as aids to particular operations, but calculation as such does not represent a very large part of the total work. It is the complex functioning and organization of the office system—the transmission of information from clerk to clerk and ensuring that it arrives at the right place at the right time—that entails so much labour and expense. The manually operated business machine can do very little about this, but the modern digital computer, with its elaborate control system designed for automatically directing and timing a long sequence of arithmetic operations, has a good deal of apparatus ready-made for such work. In particular, it is the computer's ability to " communicate " information from one arithmetic process to another, feeding temporary results into storage systems and calling on them as required, which distinguishes it from the simpler type of calculator.

A great deal of clerical labour, in factories as well as in commercial and government offices, is bound up with the task of " boiling down" vast masses of numerical data, perhaps coming from a variety of sources, to a few significant results. Normally this work might take so long that the results obtained would be completely out of date, perhaps giving a false picture of the present situation in the firm. Any policy decisions made on the basis of the results might then easily be wrong decisions. With a digital computer, however, the work of weeks or months might be reduced to hours and the decisions could be made on up-to-date information —not to mention a possible saving in labour costs.

There is actually a very wide range of jobs which the digital computer can perform in industrial and commercial offices—cost accounting, stock control, sales analysis, production control, invoicing, share yields, insurance quotations, to name just a few. One of the most popular so far has been the automatic preparation of payrolls for large numbers of employees. The usual procedure, to begin with, is that current figures from the worker's clock card—hours of work, overtime and so on—are transferred to either punched tape or punched cards. At the same time permanent information about him, like rate of pay and superannuation deduction, is " brought forward " (in the accounting sense) also in the form of punched records—some of this information having been produced by the previous week's computation.

The computer is programmed to accept the new data and to take into account all the factors which affect the total gross earnings. These might include, for example, shift allowances, variation of overtime earnings, bonuses, short time, minimum wage and sick-pay adjustments. When the gross wage has been calculated the computer goes on to income tax, insurance and the various other deductions which have to be made in order to arrive at the net wage. Finally, the machine prints a pay-slip in full detail and at the same time punches all the information to be " carried forward " in a fresh card or tape for use the following week. In addition the programme may be arranged to work out the various quantities of notes and coins needed for the actual pay. The speed at which this kind of work can be done depends on the speed of the computer and on various other factors, but it is usually in the region of 1,000 employees an hour.

The first digital computer successfully to tackle a payroll in this country was Lyons' Electronic Office, called LEO I, which was developed from the EDSAC mathematical machine at Cambridge. This computer, although now somewhat out of date (it started working in 1953), is still in regular operation at Lyons' head office, preparing the payroll for some 14,000 employees. At one time this was a full-time job for over 200 clerks, aided by conventional accounting machines. A later and improved version of the computer, called LEO II, regularly prepares the payrolls for several large outside organizations such as the Ford Motor Company Ltd., Kodak Ltd. and Tate & Lyle Ltd. LEO II is, in fact, a production model and is being manufactured and sold in quantity.

But these machines, and many others performing the same kind of work, are not normally used for payroll calculations alone. It would be quite uneconomical to install a computer for only a few hours' activity once a week and then allow it to stand idle for the rest of the time. If the machine is to justify its initial price and running costs it must be kept fully occupied—if not with the owner's work then with other people's as well, on a time-hire basis.

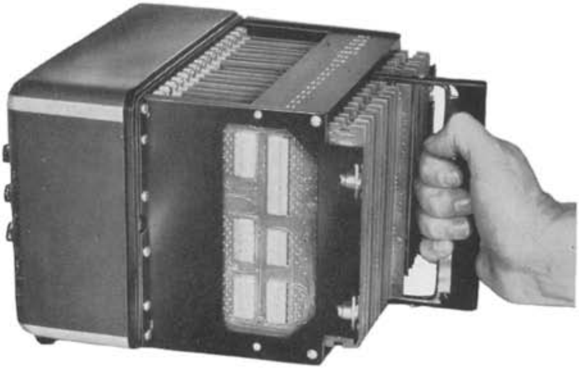
One of the biggest jobs which LEO I does between payroll programmes is the daily task of handling the orders for goods from about 150 teashops in the London area. This involves preparing all the information necessary for production in the main bakeries and kitchens, for packing and dispatching the goods, and also the records for cost accounting and management statistics. There are about 250 different items which can be ordered, so quite a large mass of data has to be handled.

The system works like this. Each shop telephones in the afternoon to say how its requirements differ from a standard order that already exists for that day. This information is recorded on punched cards, and these, together with cards carrying the standard orders, are fed into the computer. Ten shops are dealt with at a time, and the system is arranged so that while one batch of ten is being " read " into a temporary storage system the previous batch is passing through the computing circuits. At the end, an automatic printer, again working in batches of ten, prints out the necessary information in a numerical code for the packing and dispatch departments. As part of the same job the computer also does a complicated analysis on the day's figures which reveals any unexpected trends in the normal pattern of ordering. The management can then take any action which may be necessary.

Used in this particular way, LEO I can be considered as an aid to efficient production. It is not, however, performing all of the operations that come under the heading of " production control " or " production planning " in a large manufacturing concern. When a computer is applied to this kind of work the usual system is that it is fed with information on the articles to be manufactured (say the parts and materials required to make the item) and with stored reference data on the standard manufacturing methods and present capacity of the factory (for example in stocks of raw materials and machine tools available for work). It then gives out information on such things as the times at which the individual parts should be completed, and the requirements, and times of requirement, for fresh supplies of raw materials. Here the saving in clerical manpower is perhaps not so important as the saving in time. If the work of the production planning department in a factory can be cut from, say, ten weeks to two weeks on a particular job, the firm concerned might reduce its delivery time by some two months—which could make all the difference in a competitive industry.

A typical installation for this kind of work has been set up to serve the production control department of A.E.C., Ltd., of Southall, who make commercial motor vehicles. It is not a large general-purpose computer like LEO I but a medium-sized machine, with " plug-board " sequence control, used in conjunction with a fairly elaborate punched-card system. The computer as such does little more than handle the arithmetic operations, while all the other processes, such as sorting and collating, which might be done electronically in a large computer, are carried out by the ancillary punched-card equipment. This method has been chosen in preference to a completely electronic system because it is more adaptable to the particular requirements of the firm.

Plate 27 (a)—right. Airborne transistor digital computer for flight control applications made by Philco



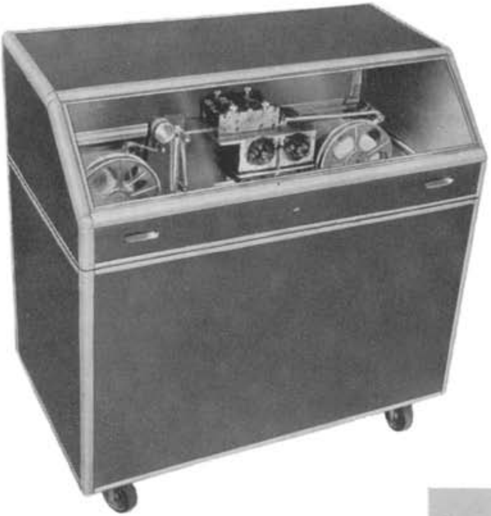
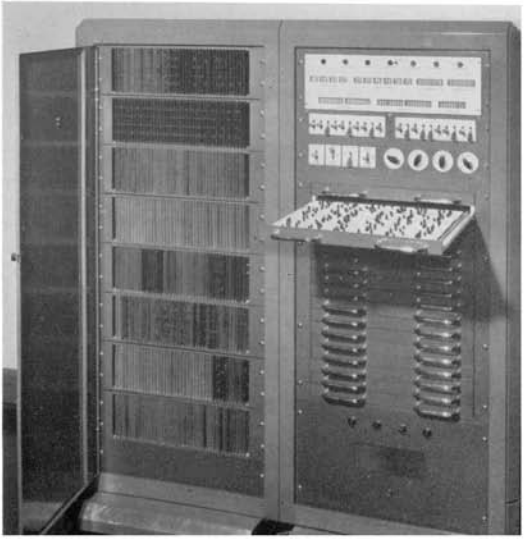


Plate 27 (b)—left. High-speed tape perforating equipment has been designed by Creed

Plate 27 (c)—right. Transistor digital computer for process control applications (Ferranti). The plug-board programme trays are clearly seen on the right



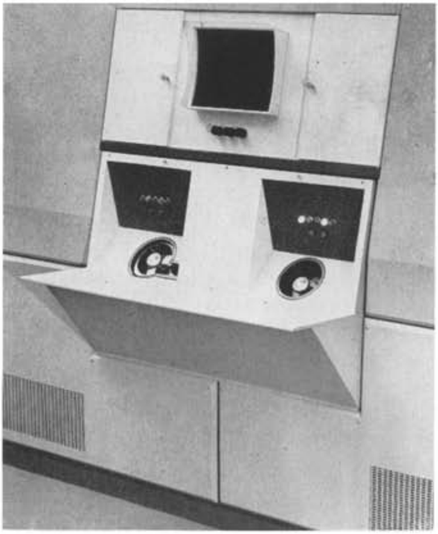


Plate 28 (a)—left. Control desk of the Solartron ERA character recognizer, showing circular magazines for inserting till rolls

Plate 28 (b)—right. Miniature magnetic matrix store by R.C.A. based on apertures in ferrite plates. Printed windings are used

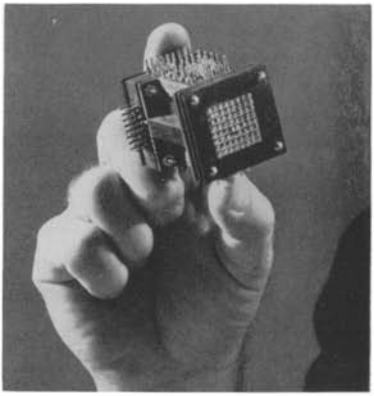


Plate 28 (c)—below. Showing some of the complex shapes that can be cut by a computer-controlled milling machine (this one using a digital control system)

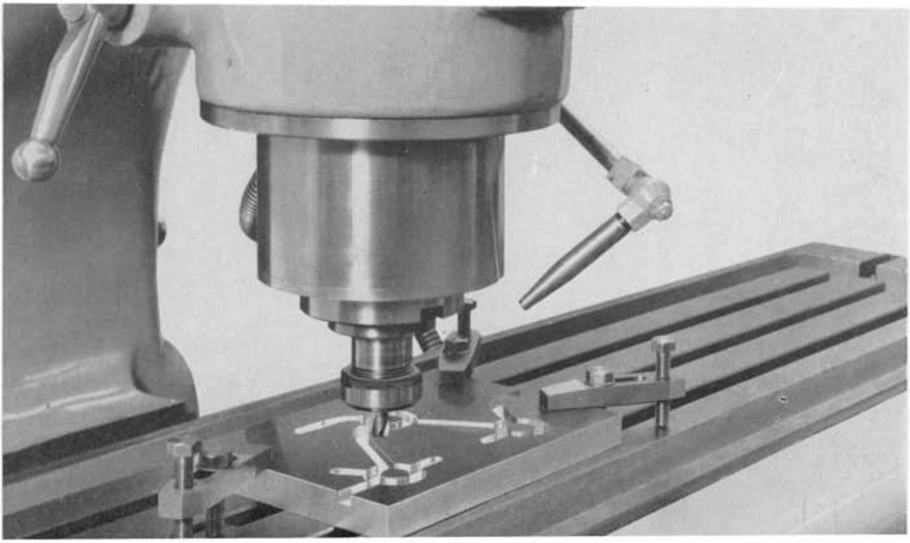


Plate 29 (a). Xerographic high-speed output printer made by Rank Precision Industries. The electronic character-forming circuits are contained in a separate cabinet

Plate 29 (b). The Metrovick 950 transistor digital computer, intended mainly for scientific work

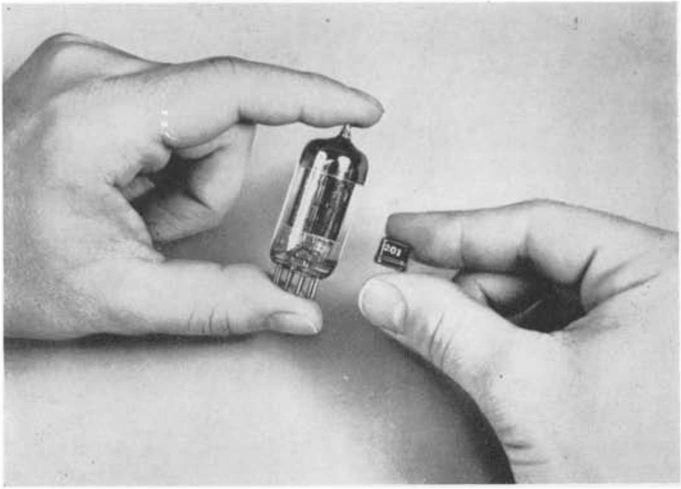


Plate 30 (a) Comparison of size between a miniature valve and a transistor. This indicates how much smaller a transistor computer can be made

Plate 30 (b). Display desk of an American military defence system based on rapid data processing

The production planning starts, of course, with the receipt of an order for a certain number of vehicles, and these might be divided amongst three, four or more different models. For each model concerned a set of punched cards is taken out of a card " library " giving details of all the different units—axles, gearboxes and so on—that go to make up that model. These are put through the computer, with other punched cards which say " 15 sets of units for Model A, 25 sets of units for Model B . . . " and so on, according to the requirements of the order. The computer then calculates the total number of units of each type required and punches blank cards with the information. Next, for each type of unit concerned a set of cards is drawn from the library describing the component parts of that unit. These are fed through the computer with the " unit cards " just punched, and the machine calculates the total number of components of each type required.

Some of these components may be already in stock, so the next job for the computer is to subtract the present stock (again recorded on punched cards) from the component requirements just calculated. It then prints the result in the form of a list of materials to be ordered, which is passed to the purchasing department. When the orders have been placed the purchasing department returns information on the quantities concerned and the names of the suppliers. This is fed into the computer by punched cards and the machine calculates in detail the rate at which the materials should be delivered to meet the production programme. The result appears in printed form, and copies of the document are sent out to the various suppliers.

Next, the computer is used to calculate the dates at which each batch of work should be started and finished in the factory, and cards punched with this information are sent to the foremen of the departments concerned. As the jobs are completed these cards are returned and checked against a time-table, and all jobs behind schedule are printed by the computer so that action can be taken about them. The computer also calculates the overall requirements for use of machine tools, or " loading ". If a particular section of the factory is overloaded, the jobs suitable for sub-contracting are automatically selected and a check is made at the same time to ensure that the jobs so affected could not be moved to an underloaded period.

Some systems of production planning use a computer to prepare a whole series of alternative production plans instead of just a single one. The individual plans can then be costed and the most economical one selected for use. Where a great many variable factors are involved, however, it is quite impractical to work out alternative plans for all possible situations. A special technique, known in economics as " linear programming ", has to be employed. This is nothing to do with the type of programming described in Chapter 11, and the technique might better be described as " optimum planning ", or finding the best possible combination of variable factors to achieve the required end.

Linear programming is not, however, confined to production planning in the narrow sense discussed above. It enables a firm to arrive at the best possible deployment of its resources—labour, transport, machining facilities, agricultural land or whatever they are—in order to achieve, say, maximum efficiency or minimum cost. Alternatively, the firm might wish to decide which contract to accept out of a number of possibles in order to make the most profitable use of these resources. The calculations involved are a matter of solving sets of linear equations in perhaps hundreds of unknowns. This can theoretically be done on desk calculating machines, but to achieve results without unreasonable delay the digital computer is essential.

It has already been suggested that calculation as such does not necessarily play an important part in data processing. Indeed, in many types of data processing there is no calculation at all and no numbers are involved. This is called " non-numerical data processing " and it may seem strange that computers are used for it. The truth of the matter is that the computer still operates on numbers inside itself, but outside the machine these numbers are only code symbols representing other types of information.

One type of application in non-numerical data processing uses the computer, and in particular its storage system, as a kind of dictionary or file of information to which reference can be made at very high speed by the electronic circuits. A notable example is the automatic translation of languages by digital computer. The simplest method of doing this is by a word-for-word translation using the main store as a " dictionary ". In the first place, of course, the words have to be coded into the form of binary numbers. All the foreign-language words likely to be required are stored in one set of addresses while their English equivalents are stored in another related set of addresses. Dealing with a particular piece of text, the foreign-language words to be translated are converted into the binary number form and fed into the computer. Each one is then compared with all the foreign-language words in the " dictionary " in turn until a correspondence is found. This is done by the arithmetic process of subtracting the incoming " number " from the stored " number ", so that correspondence is indicated when the result is zero. The generation of this zero is then used to select the appropriate English word from its related address in the store.

To achieve reliable and effective translation it is necessary to take into account the grammatical structures of the languages concerned. This involves a rather more complex type of programme for the computer, and a great deal of work on devising suitable systems has been done at Birkbeck College, London, where Dr. A. D. Booth has been an outstanding pioneer of automatic translation.

A related but slightly different data processing technique uses the store to hold information of the kind that fluctuates rapidly with changing circumstances, such as flight bookings at an airport, and where, consequently, it is very difficult to keep track of the present situation by normal methods of recording. In the business world machines specially designed for this type of work are known as file computers —because they are, in fact, mechanized filing systems. The simple computing circuits are used for " writing " the incoming information into suitable storage locations and for interrogating the record when required.

As an example, Remington-Rand have built a machine for a large mail-order firm in the U.S.A. for the purpose of keeping a running inventory of total stock and also a running total of the orders received from customers. The firm has about 8,000 items on its catalogue, while the rate of ordering might be anything from 2,000 to 15,000 orders per day. The machine actually consists of a magnetic-drum storage system, with a small arithmetic unit for addition and subtraction, and ten input units rather like desk adding machines.

Suppose that an order is received from a retailer for twenty of a particular item in the catalogue. An operator at one of the input units taps out this quantity and also the catalogue reference number on her keyboard. The magnetic-drum store is then interrogated for the catalogue number, and when this is found the recorded information on total stock and total orders appropriate to that item is transferred to a temporary storage system. Here the arithmetic unit substracts the new order of twenty from the total stock and adds it to the total of orders, after which the amended information is returned to its correct place on the drum. The whole operation takes less than half a second.

In Britain, the well-known Boots Pure Drug Company are installing an Emidec computer for the processing of branch orders. The machine maintains an inventory of all the 60,000 " lines " stocked in the firm's warehouses and shows immediately how stocks of each item fluctuate with orders from the branches. It can also give a warning automatically whenever stocks fall below a certain critical level.

The idea of using computers for the control of metal-cutting processes has already been mentioned in Chapter 6 on analogue machines. Digital computers also play an important part in this type of work, but in a somewhat different way. The movements of the machine-tool work table under the cutting tool are controlled by information recorded on magnetic tape, and it is this information which is provided by data processing in the digital computer.

A scheme devised by Ferranti is intended for manufacturing relatively small quantities of precision machined parts where the use of normal mass production techniques would be somewhat inefficient. It is based on the principle of specifying the contours of the part to be machined by a series of points, each having x and y co-ordinates from a given reference point. The x and y values are then used to move the work-table in two directions. This does not mean, however, that a human " programmer " has laboriously to put all this information on to the magnetic tape point by point. The digital computer is brought into play here, for most contours can be represented by mathematical expressions and it is only necessary to instruct the computer to calculate a straight line or a semi-circle or a parabola, as the case may be. All that the human " programmer " has to do is to feed in information about the points of change on the contours (for example, where a straight line starts to bend round into a circle) and the computer then does the necessary interpolation between them, calculating continuously the required path of the work-table in x and y co-ordinates.

The output in each ordinate is in the form of a train of pulses and each pulse corresponds to a very small unit of movement, usually one ten-thousandth of an inch. Thus the total number of pulses in each output determines the distance moved by the work-table in that ordinate, while the pulse rate determines the speed of movement. These pulses are synchronized in the x and y channels in such a manner that the work-table controlled by them follows a path corresponding to the calculated track within one ten-thousandth of an inch. Control in the vertical direction is also possible for varying the depth of cut.

One of the most vital applications of digital data processing in the dangerous modern world is the rapid handling of navigational data in military air defence systems. Here the function of the computer is to receive information from radar systems and other sources on the course, height and speed of hostile aircraft or missiles, by means of which it calculates the track of these aircraft automatically. This calculated tracking information is then used to control the launching of missiles or aircraft for the purpose of intercepting the enemy.

In the early days of radar it was not uncommon for several minutes to elapse between target detection and the passing-on of the data enabling interception to take place. This delay, of course, was due mainly to the human factor in data processing. Today, when aircraft or missiles may travel at supersonic speeds, the delay in data processing has to be cut to seconds. This can be done only by removing from the human operators all tasks involving memory and using instead the high-speed capabilities of the digital computer.

Coming now to the second group of applications of digital computers, which we have described as specialized calculations, we find the field so vast and diversified that a proper account of it would be quite outside the scope of this book. One can only look briefly here and there and get a rather kaleidoscopic picture.

To take an example, the British Transport Commission has used Lyons Electronic Office for working out distances between all possible pairs of railway stations and goods depots in Britain for the purpose of a revised freight charges scheme. The 7,000 stations and depots actually gave

50,000,000 permutations, but they were reduced to 4,000 groups to simplify the task.

In the entirely different field of biophysical research, a laboratory of London University has used a digital computer for calculating the positions of atoms in the complex molecules of living tissue. Large numbers of alternative structures have been calculated for correlation with the results of observational methods. In this way it is hoped to establish that the atoms are arranged in particular patterns which can be expressed mathematically.

In connection with the design of a multi-cored electric cable for speech transmission, the General Electric Company had the problem of determining the correct " lays " for the conductors in the cable to give the most satisfactory results. It was required to construct a cable having seven " quads ", or groups of four cores, and the conductors had to be twisted in such a way that " cross-talk " interference between them was reduced to a minimum. There are 22 different ways in which this can be done, and the best combination of seven of these had to be found. As there are 170,544 different combinations possible, each requiring, say, five minutes to work out with a manually operated calculator, the problem tackled in the normal way would take many months to solve. Using an electronic computer it took three or four days to prepare the programme and only one hour on the machine to find the solution.

This, of course, is only one of the vast number of calculations performed in the engineering field, ranging from stresses in mechanical structures to the design of nuclear reactors, from load-flow studies in electrical power generation to the critical whirling speeds of rotor systems. The aircraft industry in particular takes a large amount of computer time in calculations to do with stressing of structures, critical flutter speeds, stability in flight, heat transfer, aileron reversal, drag on wings and many other problems. In civil engineering one has heard of the more unusual job of calculating water levels at points along the River Nile over a period of 48 years—the object being to study the relative merits of irrigation and hydro-electric power schemes.

The weather forecasting services are likely to become increasingly dependent on digital computers for obtaining quick results. An entirely new method of forecasting, by calculation of the atmospheric pressure distribution for 24 hours ahead, has been developed by the British Meteorological Office, treating the movement of the atmosphere as a problem in classical hydrodynamics. To accomplish the lengthy computation in less than the three hours it takes by ordinary methods, a large digital computer has been installed at the Dunstable forecasting office.

Other applications in the field of science range from X-ray crystallography to studies of ocean tides, from astronomy to numerical analysis. There are many mathematical problems which can best be solved by numerical methods, but until the emergence of digital computers the work involved has often proved too tedious to be worth while. Now the situation is changing and there has been a great revival of interest in numerical methods of solution.

In Chapter 6 a good deal was said about the use of analogue computers as simulators. It now appears that digital computers, too, may have some applications in the field of simulation, particularly where the system to be simulated has a digital nature. As an example, the Bell Telephone Laboratories have been using digital computers to simulate proposed new coding and transmission devices for speech and television communication systems. In this way it is possible to avoid the costly and time-consuming business of constructing experimental apparatus to try out the new ideas. Digital simulation is also used for mechanical and biological systems and for testing the effectiveness of certain military tactical and defence schemes.

Finally we come to the group of applications in which the digital computer forms just one element of an overall control system. This is a fairly new field. Probably the most significant application is the control of processes in industrial plant. The idea here is that the computer is incorporated directly into the control system, taking its input in turn from various measuring transducers distributed about the industrial plant (via analogue/digital converters) and giving outputs which are used to actuate servomechanisms for controlling the behaviour of the plant.

Two machines intended for this type of work have recently been demonstrated in Britain. One, made by Ferranti, is being currently investigated as a possible means for the control of boilers during start-up and shut-down operations. The second, a National-Elliott machine, is to be used in a data processing and information system for application to a nuclear reactor of the U.K.A.E.A. Both are stored-programme computers and are quite small in physical size. The significant point is, though, that the sampling of measurement points in the industrial control system is not done so rapidly that the computing has to be performed in a continuous high-speed stream. In fact, there are pauses between each burst of computing which can be used for other calculations. As a result the machines have been designed to work in the serial mode and it is not necessary for the pulse rate, which determines the speed of computing in the serial mode, to be unduly high. Thus, although these machines are operating in what is known as " real time ", the " real time " is, in fact, rather slow.

The principle of time sharing—making use of the pauses in activity mentioned above—is likely to be a characteristic feature of the process control computer. There are often many different quantities to be measured in an industrial plant, and the computer works so fast that they can all be taken in turn, in a regular cycle of different calculations, within the normal operating time of the plant.

This situation also applies in the small transistorized computers which are now being tried out for the automatic navigation of aircraft. They are essentially automatic pilots but can cope with a great many more situations than the existing type of equipment. These might include actuating the aircraft's control surfaces to correct pitch and roll or give co-ordinated moves or required rates of climb; controlling the speed of flight; providing steering signals to fly the aircraft on a preselected path; computing the altitude for best fuel economy, greatest speed or longest range; calculating wind speed and direction for dead reckoning navigation; and storing the position and frequency of radio navigational beacons.

Information for performing these calculations is obtained by measuring and detecting devices from many different sources, including perhaps radio stations on the ground and the aircraft's own radar system. One particular computer in a military jet aircraft samples 60 inputs per second while, at the same time, computing data for 30 control outputs in an operating cycle lasting 1·8 seconds.

In such advanced applications—controlling industrial plant and flying aircraft—one feels that the computer has almost reached the ultimate point of displacing the human being altogether. But however complex the operation it performs, the computer is still only a mechanical slave, dependent on the intelligence of its human masters to frame a programme of instructions, and incapable of dealing with any situation outside of the programme. Whether in the future we are likely to have computers with capabilities approaching perhaps the lower levels of animal intelligence is a subject which is best left for discussion in the final chapter.

13

RECENT DEVELOPMENTS

When thermionic valves were introduced into computers they made possible a greater flexibility of design and also an increase in the speed of operation. But, admirable as they are, valves have one or two drawbacks which become apparent when several thousands of them are massed together in a large machine. First of all they take up a good deal of space, not being very much smaller individually than the electro-mechanical relays used in earlier times. Secondly, they consume large quantities of electric power—perhaps 10 kilowatts or so in a large machine—and thirdly, arising out of this, they produce a considerable amount of heat, which may endanger the reliability of other components. Finally, they have a certain rate of failure and limitation of life which, although hardly perceptible in a domestic radio receiver, may have a considerable nuisance value in a large computer.

In the next few years, however, it is likely that we shall see the gradual disappearance of the thermionic valve from computers. There are several new devices which seem to be making a fair bid to take its place, in digital machines at any rate. Of these the most important is the transistor. This tiny device (see p. 206, Plate 30 [a]) works on the diffusion of electric current carriers through solid semiconductor materials such as germanium and silicon. From the functional point of view it is roughly comparable with the triode valve (see Fig. 13.1). The emitter electrode can take the place of the valve's cathode in electronic circuits, the collector electrode can take the place of the valve's anode, while the base electrode can perform a similar controlling function to the valve's grid. (The rôles of the emitter and base, however, are to some extent interchangeable.)

Apart from its small size, the transistor is quite robust, produces very little heat, is efficient in operation and appears to have a long expectation of life (so far as we can tell at present). In addition it will operate from a single source of power of only a few volts. All these features make it an ideal component for use in digital computers, particularly in applications where size and weight are important. The small transistor computer in Plate 27 (a), p. 203, for example, is intended for use in aircraft control systems.

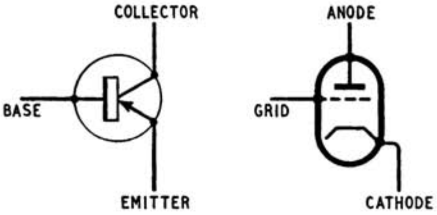


Fig. 13.1. The graphical symbol of a transistor (extreme left), for functional comparison with that of a valve (left)

Some of the first transistor bi-stable circuits made use of the negative resistance characteristic of the point transistor to give a regenerative action which produced a rapid snap-over from one stable state to another. In this way it was possible to make one transistor do the work of two valves (or a double valve) in an equivalent " thermionic " computer—which, of course, was a considerable advantage.

Unfortunately, the point transistor proved to be somewhat unreliable. Apart from its temperature sensitivity, high noise value, general fragility and liability to burn out, it had characteristics which varied widely from unit to unit and this made it difficult to design two-state circuits with consistent " on " and " off " conditions. The real future lies with the junction transistor, which is a much more reliable component and is now manufactured in large enough quantities to make it commercially practical for such applications.

One of the big drawbacks of the conventional junction transistor, however, is the inherent slowness of the current carriers diffusing through the base material, compared with the speed of the electrons in the valve. With the audio-frequency type of junction transistors this limits the pulse repetition frequency to something in the region of 50 to 100 kc/s. Thermionic-valve computers, on the other hand, commonly work with p.r.fs of several megacycles. However, a good deal of work has been done on the development of high-frequency junction transistors with alpha cut-off frequencies in tens of megacycles. These are now becoming available in sufficient quantities to allow complete computers to be built with p.r.fs comparable with, and even higher than, those of the thermionic-valve machines.

Some of the latest transistor machines to be developed are using high-frequency transistors of the alloyed-junction type and these are making possible pulse rates in the range 100 kc/s to 1 Mc/s. As an example, Ferranti have recently developed a transistor computer with a pulse recurrence frequency of 500 kc/s (see p. 203, Plate 27 [c]) for use in the new sphere of process control by computers, which has already been mentioned in Chapter 12. The machine performs addition, subtraction and other simple operations, such as transfer of numbers, in 24 μsec. Logical circuitry is similar to that in the larger " Pegasus " valve machine made by Ferranti. It is based on " OR " and " NOT " gate-type circuits formed by semiconductor diodes and transistor inverters. The two-state elements which provide short-term storage registers are formed by transistor Eccles–Jordan trigger circuits. Pulse rise times are about 50 μsec. Power transistors are used for driving the magnetic-core matrix store, as quite heavy currents are required for this purpose. The computer works on supplies of +6 V, −6 V and −24 V and its total consumption is only 250 W.

A particular kind of high-frequency transistor now coming into wider use for computing circuits is the surface barrier type. This has a very narrow base region which reduces considerably the transit time of the current carriers and makes possible p.r.fs of anything up to 8 Mc/s. Moreover the characteristics are such that the range of base voltages required for satisfactory on-off operation comes within that normally provided by the collector. This means that the collector of one transistor can be connected directly to the base of another transistor without any intermediate components or biasing arrangements. As a result a very simple kind of circuitry can be obtained which is extremely economical in components. Only resistors are required, in the ratio of about one to every four transistors.

An example of a direct-coupled trigger circuit using such transistors is shown in Fig. 13.2 (a) and an " OR " gate at (b). These can be compared with the equivalent valve circuits in Chapter 7. In the trigger circuit (a), if sufficient current is applied to the base of V_1 to drive the transistor

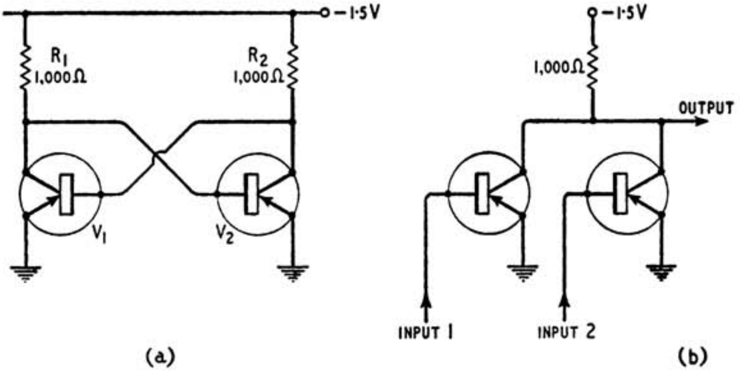


Fig. 13.2. Transistor trigger circuit comparable with the Eccles-Jordan valve trigger circuit at (a); transistor " OR " circuit at (b)

into saturation, the collector voltage is low enough that when applied to the base of V_2 it cuts off the second transistor. This causes the collector voltage of V_2 to be high (in a negative direction), and sufficient current flows through R_2 to the base of V_1 to maintain the saturated condition of V_1. This provides one stable state of the circuit. The other stable state is with V_1 cut-off and V_2 saturated. Triggering from one state to the other is achieved by applying currents from suitable voltage pulses to the bases of the transistors—negative-going or positive-going as appropriate. The voltage change obtained at the collectors between the " on " (saturated) state and the " off " state is about $-0{\cdot}03$ to $-0{\cdot}35$ V, roughly ten times.

In the " OR " circuit at (b), currents from negative-going input digit pulses are applied to the bases of the transistors. Each of the transistors must be capable of allowing sufficient current to pass through the load resistor to drop the output voltage to near earth potential. This positive-going output pulse could, for example, be applied directly to the base of the " on " transistor in the trigger circuit (a) to reverse its state.

Similar direct-coupled circuits can be obtained with other types of transistor, but the surface barrier transistor offers the additional advantage of high speed operation.

A good many types of logical circuits have been devised for transistor computers. Quite a common arrangement is to use semiconductor diodes for the logical gating operations and the transistors merely as pulse amplifiers. One system known as " ballot box logic " uses the transistors in this way and employs small transformers with various input windings as the combining elements. Yet another technique depends on rectangular-hysteresis-loop magnetic cores (as described in Chapter 9) to provide the stable states and uses the transistors to drive them from one state to the other.

Magnetic cores have been under investigation as possible computing elements for some time, not only for storage but for arithmetic and control operations as well. Like transistors they are small and robust, long lived, efficient in operation and produce very little heat. The rectangular-loop core has the important advantage that it will remain in either of the two states of maximum remanent flux without a continued dissipation of power (unlike valve and transistor circuits). Its big disadvantage, however, is that, because of the time constant of the windings on the core, it will not operate at high pulse repetition frequencies (above about 200 kc/s) and this is one reason why it has not been widely adopted in computers except for storage purposes.

On the other hand, this limitation does not matter for computers which are required to operate only at low speeds. A recent example is a process-control computer designed by National-Elliott (p. 30, Plate 6), in which the digit pulse rate is 166·5 kc/s. Addition, subtraction and 46 other orders are performed in 612 μsec. each. This computer is actually based on the technique of combining magnetic cores with transistors, mentioned above. The logical circuitry is constructed from " OR " and " NOT " elements, and for this the two-state magnetic ferrite cores have various sets of input windings and a single output winding. According to the logical pattern of input digit pulses applied to the input windings, each core is set into one state or the other, " 0 " or " 1 ", by the coincident magnetic fields. This happens in the first 3-μsec half of the 6-μsec digit pulse period.

A so-called " shift " or " trigger " pulse is then applied to an additional input winding in the second 3-μsec half of the digit-pulse period, and the effect of this is either to switch the core or leave it as it is—depending on how it has been set by the original digit pulse inputs. As a result of this action the output winding either produces a pulse or a space (absence of a pulse), and this is where the transistor enters the picture—to amplify, regenerate and pass on the output pulse to another winding on another core.

Thus the magnetic-core/transistor circuits operate in two phases—the " set " phase, when the cores are switched into one state or the other, 0 or 1, by the input signals, and the " shift " phase, when the information on this state appears at the output. The action is of necessity slower than in diode logical circuits, where usually all the switching happens at once, and in any case it is limited by the time constant of the windings mentioned above.

The speed of operation of magnetic-core/transistor computing circuits can be increased by working in the parallel mode instead of the serial mode. As explained earlier, this means that all the digits of, say, a 32-digit number appear as pulses or spaces on 32 wires simultaneously, instead of in a time sequence on just one wire. Such a technique has been adopted in an Emidec digital computer built by E.M.I., a somewhat larger machine than those mentioned previously. This uses magnetic cores in combination with a.f. transistors and high-frequency types. It has a high-speed " immediate access " magnetic-core matrix store driven by power transistors, together with supplementary magnetic-drum stores.

Although the basic pulse rate of the E.M.I. computer is even lower than that of the National-Elliott (actually 112 kc/s) the speed of computation, owing to the parallel mode of operation, is a good deal faster. Addition, subtraction and other similar operations take about 120 μsec, in fact. This speed is necessary, however, where it is not in the National-Elliott, because the E.M.I. computer is intended for entirely different applications. Moreover, it has to be paid for in the greater complexity of equipment necessary with parallel operation. The size of the E.M.I. machine is considerably smaller than that of equivalent valve machines and the power consumption of the computer proper is only 2 kW—incidentally it runs off car batteries!

E.M.I. have also developed a very much faster machine which uses diodes and h.f. transistors for the logical circuits and operates at the high pulse rate of 1 Mc/s. Magnetic cores are only present in the matrix store. Another quick access store (64 words capacity) is formed by combinations of capacitors and diodes. Instructions are of the two-address type and again the machine operates in the parallel mode. The average time for addition and similar operations in this computer is only 0·6 μsec—about 200 times faster than the magnetic-core/transistor machine.

A good deal of work is being done on the development of transistor computers for operating at high pulse rates in the region of 10 Mc/s. Micro-alloy and diffused-base transistors are used for this work and will operate at pulse frequencies in the 20–50 Mc/s region. At these high pulse rates transistors have an extra advantage over the thermionic valve, apart from those mentioned on p. 217, for they are inherently low-impedance devices and therefore reduce the effect of circuit capacitances on the rise and fall times of the pulses. It is important, too, that transistors allow the equipment to be made quite small, because at high pulse rates the time delays introduced by the normal conduction of electricity along the wiring could upset the precise timing arrangements which are so essential to the correct operation of the computer.

This problem is particularly important in the design of future computers which will have information transfer speeds of the order of 10^9 binary digits per second (as compared with the present 10^6 bits/sec). For example, it takes an electrical signal 1 mμsec (millimicrosecond) to travel along about 6–8 in. of wire. This means in practice that future computers working with mμsec pulse intervals cannot be much more than about 2 ft^3 in size if the timing problem is not to become serious.

It is clearly impracticable to compress thousands of conventional transistors and other components into a 2-ft cube by the normal methods of electronic assembly. New methods of fabrication are therefore being developed by which whole sections of a computer can be " printed " by systems of chemical processing. One promising approach has become known as " solid circuitry ". In this, complete integrated transistor circuits are produced by depositing films of various materials on to small wafers of semiconductor material and selectively etching them away to form resistive and capacitive elements connected by conductors to the semiconductor transistor or diode elements. These transistor or diode structures are formed by initially diffusing impurities into the semiconductor material.

An example of a single solid circuit compared in size with its conventional counterpart, is shown in Plate 31 (b), p. 231. It is a two-state circuit, containing as integral parts of the semiconductor material the equivalent of twelve normal electronic components—two diffused-base transistors, two capacitors and eight resistors. It measures less than $\frac{1}{4}$ in. × $\frac{1}{8}$ in. × $\frac{1}{32}$ in. Related techniques are being developed for " printing " whole arrays of such circuits in close formation. Component densities of the order of 34 million per cubic foot should soon become possible by this means.

Apart from transistors there are several other semiconductor devices which have interesting possibilities. The double-base diode, for example, has a negative resistance characteristic rather like a point transistor which makes possible a regenerative switch-over action. As a result it will do the work of the two ordinary transistors required for a bi-stable circuit. A negative resistance characteristic is also possessed by a new German switching transistor. This is a junction type with a base of normal thickness, but has a tungsten whisker embedded in the collector so that it just penetrates the collector junction. The electric field that is set up in the base region by the whisker accelerates the current carriers and by this means gives a short switching time which allows pulse repetition rates of up to 2 Mc/s.

Another new semiconductor device which makes possible two stable states in one unit is a special silicon diode composed of four layers. The two states are an " open " or non-conducting state of about 10 MΩ and a " closed " or conducting state of about 20 Ω, and switching from one to the other is effected by the voltage applied. If this voltage is made to exceed a certain critical " breakdown " value the diode is switched from the " open " state to the " closed " state, and remains closed provided there is sufficient current to hold it there. If the current is reduced below this " holding " value the diode switches back to the " open " state again. A typical value for the breakdown voltage is 30 V, with a current of several hundred microamps, while the voltage required to maintain the minimum holding current is about 1 to 2 V. In the future, the " transistor diode ", as it is sometimes called, is likely to be outstanding for its large power handling capacity compared with the conventional transistor.

A " breakdown " action of similar nature occurs in the avalanche junction diode, which utilizes a current-multiplying effect rather like the sudden ionization in a gas discharge tube. Normally, junction diodes have a slow response when they are rapidly switched from forward voltage to reverse voltage, because storage of current carriers prevents the diode current from immediately falling to its normal reverse-voltage value. A certain recovery time is necessary, which sets a limit to the operating p.r.f. The avalanche diode, however, works with a reverse bias just short of the breakdown voltage, instead of at zero voltage, and the applied switching pulse initiates the avalanche effect.

One of the most interesting computing elements recently developed is the cryotron. It is based on the phenomenon of superconductivity, or zero electrical resistance, which occurs in metals at extremely low temperatures. The device takes the form of a short wire " core ", perhaps an inch long,

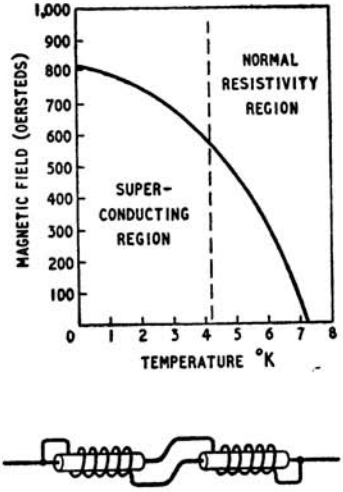


Fig. 13.3. Graph shows how the transition from superconductivity to normal resistivity is controlled by the magnetic field in cryogenic devices. Below is a two-state element using two cryotrons

carrying a small coil, and it has the advantages of extremely small size, simple construction and low power dissipation.

In superconductivity, the resistance of a metal suddenly drops to zero when its temperature is reduced below a certain critical value (by liquid helium cooling). The presence of a magnetic field makes this critical temperature even lower, and Fig. 13.3 shows how a field affects the onset of superconductivity for a " core " made of lead. Thus, a wire held at a constant low temperature can be switched between superconductivity and normal resistivity (up and down the dotted line) by means of an energizing coil—thereby providing a two-state device suitable for binary arithmetic.

A two-state circuit using two cryotrons is shown in the lower diagram. If a current is established in one wire " core " it passes through the coil of the other and drives the second " core " into normal resistivity—the current continuing indefinitely because the energizing coils themselves are held permanently superconductive. An extra pair of cryotrons puts the circuit into the desired state.

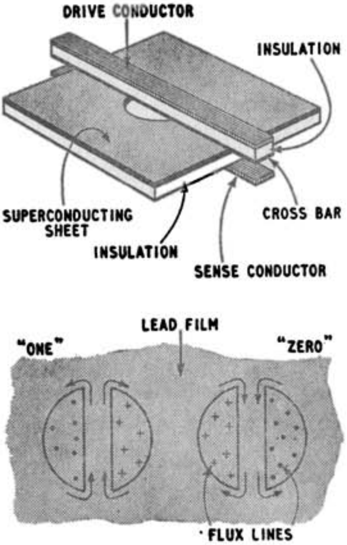


Fig. 13.4. Superconductive two-state device based on circulating currents in a lead sheet. Above is shown the form of construction, below the path of the currents

Another superconductive two-state device has been developed which is somewhat different in principle and is said to be much faster in operation than the cryotron. This depends on the magnetic flux produced by circulating currents induced in a superconductive lead sheet. (The superconducting condition being obtained by operation at temperatures below 10° K.) The lead film, deposited on an insulator, has a hole cut in it with a lead bar metallized across as shown in Fig. 13.4. When a current pulse is sent through the drive conductor the resultant build-up of magnetic flux links with the superconductor and induces currents in it. These circulate indefinitely because of the zero resistance and set up their own magnetic flux. Whether a "1" or a "0" digit is stored is determined by the direction of the induced currents.

Actually, the initial build-up of induced current is quite complex because the presence of a magnetic field affects the threshold of superconductivity and the induced magnetic field opposes the driving field. Reading-out is achieved by sending a current in the reverse direction along the drive conductor. This causes the induced currents to collapse, and the resultant change of magnetic flux induces a current pulse of one direction or the other in the sense conductor.

Probably the most advanced programme of research into superconductive computer elements has the ultimate aim of " printing " whole arrays of tiny cryotrons in closely packed formation—each element being small enough to fit into a 1-μ square. Conductors will have to be about 0·1 μ in thickness. This work again is an attempt to achieve the small dimensions for computers which will become necessary for operating speeds of 10^9 bits/sec. Such speeds are likely to be very necessary for advanced future machines of the type which " learn " by experience (to be described in the next chapter), since these self-adaptive computers have to execute a great many trial calculations at high speed before optimum results are produced.

Materials with rectangular hysteresis loops (as described in Chapter 9) are still being investigated for storage purposes and new devices are being developed from them. One interesting example is the magnetic-cell storage system. This consists of a block of ferrite with small holes made in it which have wires threaded through them. Each hole and its surrounding material (a magnetic cell) is equivalent to a small ferrite core and the threaded wires form reading and writing windings. The system has the advantage that it can be switched by smaller currents than are required for separate ferrite cores and can therefore be driven by transistors. Its main disadvantage is that it does not saturate. This means that the stored flux is largely dependent on the applied current, so that the two states are not so well defined and uncritical as in the separate ferrite core.

One development of the magnetic cell idea has been to use printed conductors for some of the " windings " instead of threaded wires. This is possible because the ferrite storage medium is a ceramic-type material and therefore an insulator. As a result it is possible to make an extremely compact matrix store equivalent to the core types in Chapter 9 but considerably smaller.

The example in Plate 28 (b), p. 204, will store 2,560 binary digits in a volume of only 2 cu. in. It is made up of thin plates of ferrite, each containing 256 holes of 0·025 in. diameter arrayed in a 16 × 16 matrix on centres spaced by 0·05 in. A printed " winding " threading all the holes is formed by metallizing conductors on both sides of the plate and on the walls of the holes. When the plates are stacked one above the other all the holes are in accurate alignment, and ordinary wires are passed through them to form other " windings ". The printed conductors serve for writing and reading purposes and the ordinary wires for address selection.

A new kind of square-loop ferrite store developed by Mullard is intended for computer speeds up to ten times faster than those possible with the existing types of magnetic matrix stores. It achieves the higher speed of operation by avoiding the need to switch the cores into their full states of saturation. Two ferrite cores are required for each digit stored. They are switched in opposite directions of magnetization by a wire carrying the input digit pulse and, coincidentally, in the same direction by a read/write drive wire. Thus one core is magnetized more than the other—which one depends on whether a " 1 " or a " 0 " is stored. The output wire threading the two cores is arranged to give " windings " in series opposition (like the input wire). When the read/write drive wire is energized to " read ", the differing magnetic flux changes in the two cores induce in the output wire a combined output current pulse, the direction of which depends on whether a " 1 " or a " 0 " was stored.

An entirely different magnetic storage principle which may lead to simpler and cheaper matrix stores than the core or cell types is based on helical magnetization patterns

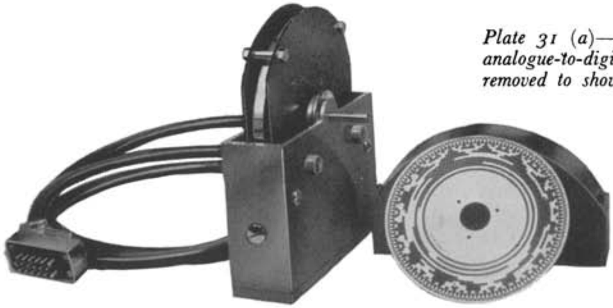


Plate 31 (a)—left. Electromechanical analogue-to-digital converter with the scale removed to show the binary system of coding

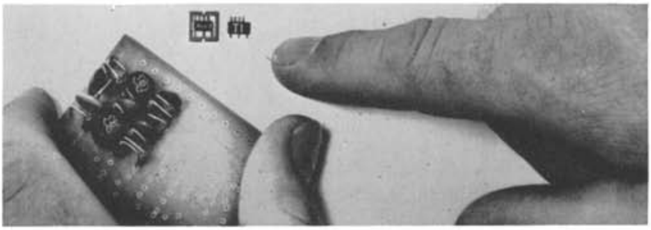
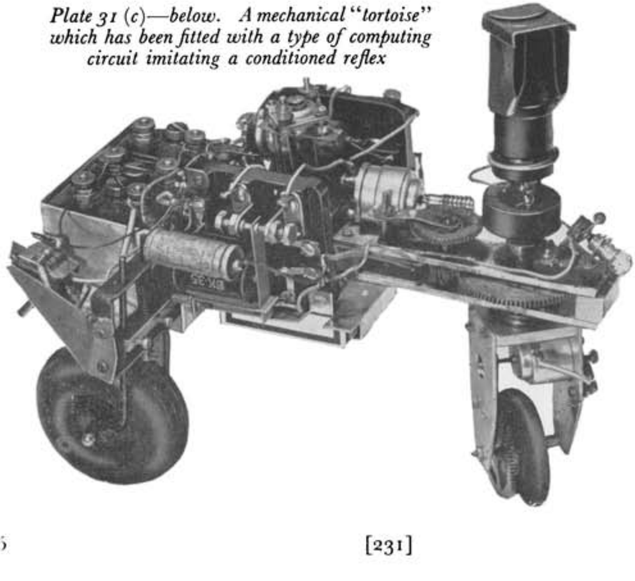


Plate 31 (b)—above. Semiconductor solid circuits (pointed to by the finger) compared in size with a circuit using conventional components

Plate 31 (c)—below. A mechanical "tortoise" which has been fitted with a type of computing circuit imitating a conditioned reflex



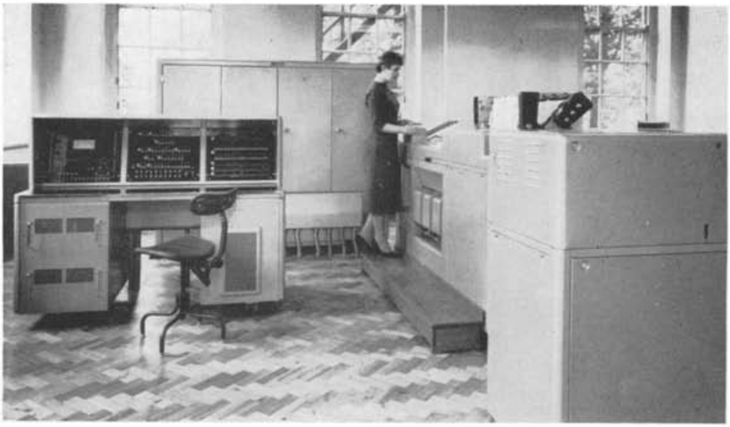
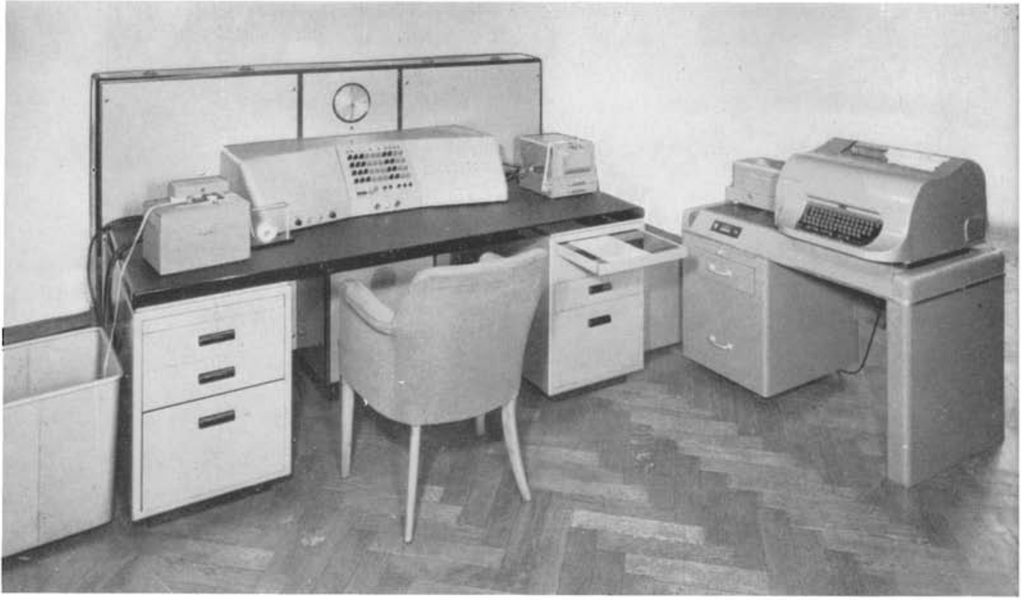


Plate 32 (a)—above. Digital computer manufactured by International Computers and Tabulators in use by British Railways. Plate 32 (b)—below. A desk-size transistor digital computer designed by Ferranti



in magnetic wires, produced by the application of coincident circular and longitudinal fields. The matrix consists of arrays of vertical magnetic wires interwoven with horizontal copper wires. Current passed through the magnetic wires produces the circular fields around them and current through the copper wires the longitudinal fields.

The preferred direction of magnetization in the magnetic wires is altered from the normal longitudinal path to a helical path by mechanical torsion. The storing of a binary digit requires two coincident current pulses—one in a magnetic wire and the other in a copper wire. Reading out is accomplished by applying a strong longitudinal field in the reverse direction, and the read-out signal is detected across the magnetic wire. Transistors have been used for the drive circuits, working at pulse rates of about 500 kc/s.

Probably the most recent development in magnetic storage systems is the experimental use of thin magnetic films as the storage medium. Small spots of magnetic material, a few millimetres in diameter and up to 1000 Ångströms thick, are deposited in regular arrays on to a glass base. Each spot acts in much the same way as a ferrite toroidal core in the present matrix type of store (see p. 135). The material has a rectangular hysteresis loop and it can be switched from one direction of magnetization to the other by currents passing through adjacent conductors, which can be printed on both sides of the glass base. Coincident-current methods can be used for the driving system, as in the present magnetic matrix stores.

This technique makes possible very large storage densities—perhaps of the order of 1,000 spots per square centimetre. In addition, the magnetic film has the great advantage of a low switching coefficient—defined as the product of switching time and applied field. This coefficient is at least ten times smaller than the value for ferrite toroids, so in general it is possible to obtain much faster switching times (in the range of 1–10 mμsec) and use much smaller driving currents. Transistors are therefore quite suitable for energizing the conductors.

The possibility of using transistors for driving purposes also occurs with ferroelectric stores based on a new material called triglycine sulphate. This has the same kind of rectangular voltage hysteresis loop as the well-known barium titanate but with a much lower coercive field, allowing it to be switched from one state to the other with potentials of only about 20 V. Repeated switching does not cause any fatigue in this material, as it does in barium titanate, and a given area will retain a given polarization indefinitely without deterioration. Switching times of the order of 1 to 2 μsec can easily be attained. A matrix store can be constructed with 30 or more electrodes to the inch evaporated on to each side of a thin slice of crystal.

The problem of time delays in extra high speed computers can be tackled in another way apart from that of size reduction. That is, to use lengths of conductors which are precisely related to the phase of the signals—in other words, transmission lines. This, in fact, is now being done in experiments on a new class of parametric-oscillator computing circuits working at extremely high radio frequencies of the order of 2,000 Mc/s.

The parametric oscillator has already been widely exploited in Japan as a binary computing element—called the parametron—working at radio frequencies up to about 3 Mc/s. The " 0 " and " 1 " states are expressed as two possible phases, 180° apart, into which the oscillation can be locked by input trigger signals. The parametron requires no valves or transistors, only passive reactive elements, and being therefore extremely stable, reliable and long-lived, is ideally suited for use in digital computers. The main limitation is that because several cycles of oscillation are required to establish a binary digit, " 0 " or " 1 ", the digit rate is necessarily low (up to 150 kc/s) with the 3 Mc/s oscillation frequency.

By increasing the oscillation frequency, however, the computing speed can be greatly improved; but it is necessary to go up to several thousand megacycles to obtain really high speeds. Hence the need for transmission lines. The experimental parametric-oscillator circuits mentioned before have been constructed from strip transmission lines by " printing " techniques and use integral semiconductor diodes as variable reactance elements. With oscillation frequencies of 2,000 Mc/s it is possible to obtain binary digit rates of up to 4×10^8 per second. It may be possible, by developing the technique, to achieve rates as high as 10^9 binary digits per second.

A section of the digital computer which calls for considerable engineering development is the input and output equipment. As already mentioned in Chapter 10, the main limitation of the existing type of output equipment is that it cannot work at the " electronic speed " of the computing circuits and therefore produces a " bottleneck " in the operation of the computer as a whole. Then, of course, there is another " bottleneck " in the time required for the human operators to translate numerical data from ordinary documents into the punched-card or tape form required by the input mechanism—and back again at the output end.

The ideal arrangement is obviously a computing installation which will accept numerals or letters as they are printed or typed on ordinary documents and then produce the results of the computation in similar form. A great deal of development work is being directed to this end, and it is impossible to deal with all of it. Some idea of the problems involved, however, can be gained from the design of an automatic character " recognizer " which has been built by Solartron for translating numerals and other characters either into punched-card or tape form or direct into digit pulses for feeding into a computer.

A good many character recognizers have been devised in recent years and in all cases the main difficulty has been in making them flexible enough to cope with different character styles, imperfections in printing, spots of dirt on the paper and so on, at the same time as achieving complete reliability of recognition. The general principle is that the printed characters are fed in turn past a scanning system, either electronic or electro-mechanical, which converts each one into an electrical pattern corresponding to the disposition of light and dark on the paper. This electrical pattern is then compared with reference patterns, representing the complete range of possible characters, stored inside the machine. When a correspondence is detected, a signal indicating the appropriate character (perhaps in binary or some other code) appears at the output.

One recent design trend is towards the use of logical circuits of the kind found in digital computers—notably " AND " and " OR " gates. If the characters are broken down into small black and white areas, or " picture elements ", it is possible to identify a character by the fact that certain elements are either predominantly black or predominantly white. There are only two conditions, corresponding to " on " and " off " in electrical form, and this is what makes it possible to use the logical two-state gating circuits.

The electronic circuits in these " logical " machines, then, have to work on statements of identification something like this: " If black is either in area B5 *and* area F2 *or* area D3 *and* area 15, *and* is in area F6 *and* area H5, but is not in area E4 *or* area G3, then the character is ' 4 '." The *ands* and *ors*, of course, are instrumented by gate circuits performing these functions, and it is the interconnections of the gates which form the reference patterns against which the incoming characters are tested.

This kind of principle is, in fact, used in the Solartron machine, which can recognize numerals at rates up to 300 per second. It has to be programmed for different typefounts, but allows for all the variations of pattern caused by

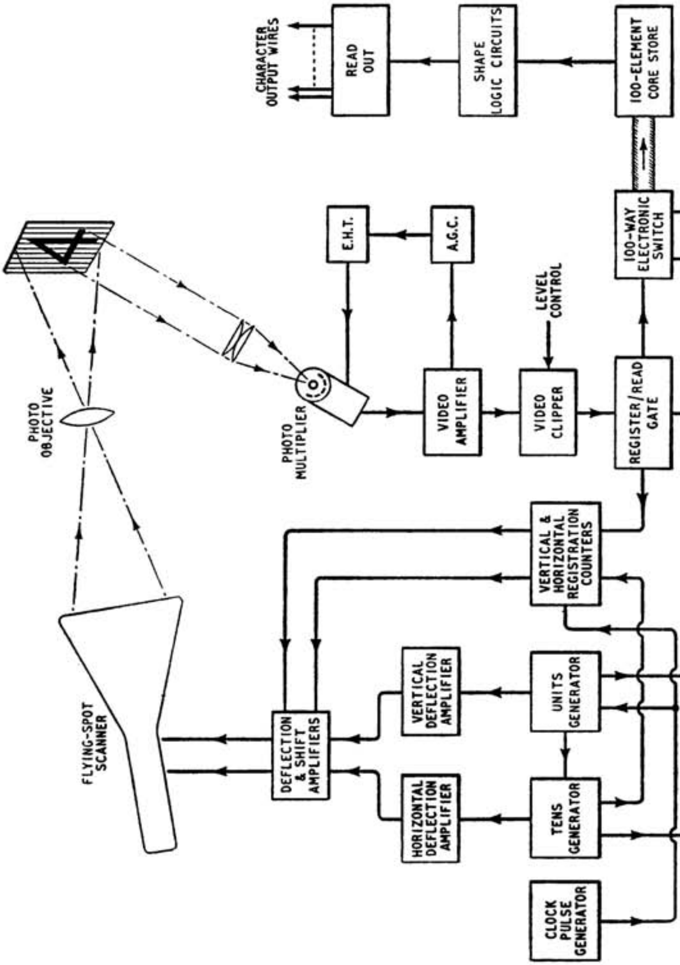


Fig. 13.5. Schematic diagram of an automatic character recognizer

bad or misregistered type, smudging, dirt on the paper, and so on which one expects to find on ordinary business documents.

A photograph of the control desk of the Solartron machine appears in Plate 28 (a), p. 204; while a block diagram of this type of machine is shown in Fig. 13.5. A paper feed mechanism moves the characters in turn past a flying-spot scanner, and the variations in reflected light resulting from the scanning process are picked up by a photo-multiplier tube. The signals from this are amplified and then passed through limiting circuits which clip the top and bottom of the waveform.

This establishes definite black and white levels which are independent of the heaviness of the printing, the condition of the paper and so on.

The actual scanning is done in ten vertical lines, and each line can be considered as being divided into 10 "picture elements". Thus there are 100 picture elements altogether, each of which will be either predominantly black or predominantly white. (Other numbers of scanning lines and picture elements are used in different versions of the machine, but a 100-element system will be assumed here for the sake of explanation.) The scanning speed is synchronized by the clock pulse generator (of 500 kc/s p.r.f.). The output of this is frequency divided down to give a line scan period corresponding to 10 picture-element periods (the "units generator") which is used to synchronize the vertical deflection. Another frequency divider (the "tens generator") then gives a signal for synchronizing the horizontal deflection.

There are actually two scanning cycles for each printed character. The first is used for automatic registration of the scanning raster on to the centre of the character, while the second is the actual reading period. This accounts for the "register/read gate" which, under the control of the timing generators, causes the photo-multiplier output to be switched either to the registration system or to the 100-element magnetic-core store. The distribution of the black and white picture elements to corresponding positions in this store is done by a 100-way gating system under the control of the " units " and " tens " generators.

From the store, the " on-off " pattern representing the read character is applied to the logical recognizing circuits, which work on the kind of principles mentioned before. The actual arrangements of gates are extremely complicated, but in general terms the initial identification processes are performed by groups of " OR " gates which, by their nature, allow for certain possibilities and variations in the patterns. The really conclusive and final decisions are then made by groups of " AND " gates, which are fed from the outputs of the " OR " gates. Semiconductor diodes are used for these " OR " and " AND " circuits, and the interconnections between the gates (i.e., the reference patterns determining which paths the input information shall take) are set up, or programmed, by printed circuit boards.

At the end of the recognition process for each character a signal appears on one of the output wires (representing the numerals or other characters) and this can be used for operating, say, a card punch or the keyboard of an automatic typewriter. Alternatively, the output signals can be fed directly into a digital computer after conversion into binary-coded decimal digits. One version of the machine will identify the numbers 0 to 11, the fraction ½, the plus sign and eight specified letters.

The registration or centring process mentioned earlier is achieved by detecting the position of the top, bottom, left and right extremes of the character in the initial " register " scan, then applying appropriate shift voltages to the scanning tube during the " read " scan. For horizontal registration, a counter is used to count the number of vertical scan lines from the left to the first and last elements which have any black content. Half of the difference between these two counts (i.e., the centre of the misregistered character) then gives a measure of the shift from centre required to be applied to the existing scan. For vertical registration a similar process is used, working with numbers of elements instead of lines. Any progressive tendency in character misalignment is corrected by a servo system on the paper feed mechanism.

Recent research on character recognition tends towards the idea of working on the shape or outline of the characters—a morphological rather than a geometrical approach. This does not depend even on the characters being the right way up, since their shapes remain the same whatever position they occupy.

Coming now to high-speed output systems, the most advanced idea for obtaining the numerals directly written on paper makes use of a cathode-ray tube which " writes " them on its screen in rapid succession. The images are then recorded photographically on paper by the dry xerographic process. This dispenses altogether with the comparatively slow mechanical methods of making marks on paper. A machine working on this principle called the " Xeronic " page printer has been developed by Rank Precision Instruments. It can print at the staggeringly high speed of 1,500 lines a minute with 128 characters in each line on 13-inch wide paper, and is being developed to work at up to 3,000 lines a minute.

The machine is a combination of the " Copyflo " (Plate 29 [a], p. 205), an existing xerographic copying machine made by Rank-Xerox, Ltd., and an electronic character-generating equipment housed in a separate cabinet (except for the c.r. tube).

Special cathode-ray tubes have been developed in the U.S.A. for forming characters by passing the electron beam through " stencils " and focusing the resulting patterns on the screen, but these are rather expensive items. The interesting point about the Xeronic machine is that it uses a normal type of c.r. tube and the characters are formed rather like Lissajous figures by applying deflection signals to the x and y plates. These deflection signals are produced by applying suitable waveforms to passive networks and the results displayed on the screen (capital letters) are of excellent quality and clarity. Transistors are used extensively throughout the machine.

Other electronic circuits are used to decode the binary input signals which specify the required characters and also to decode input instructions relating to the layout of the printed data. These layout instructions include such functions as horizontal and vertical tabulation, line feed (equivalent to carriage return on a typewriter), spacing and stopping, and selection of forms and headings as may be required for bills, invoices and so on. Tabulation can be pre-set manually by a plug-board, while the size of the characters printed can be varied by adjusting the gain controls of the deflection amplifiers.

The actual xerographic process of recording the characters displayed on the c.r. tube is centred around a large rotating drum with a coating of selenium. To begin with the drum surface receives an electrostatic charge from a wire grid, and this is retained as long as the drum is in darkness. Next the images from the c.r. tube are focused on the drum, and these modify the charge on the selenium to form a charge pattern corresponding to the characters displayed. The resultant electrostatic image is " developed " by showering over the drum thermoplastic powder which has been charged oppositely to the pattern of characters on the selenium so that it adheres to the electrostatic image. After this the powder image is transferred by application of another electrostatic charge to the band of 13-inch paper, which is in continuous contact with the face of the drum and moving at the same speed. Finally the powder image is fused permanently to the paper by passing under a heating element. At the end of the process the drum is cleaned of powder and is then ready for further recording action.

Developments in analogue computing have not been so spectacular as in the digital field but are none the less real. New circuits are being devised, using hitherto untried components like transistors and magnetic cores, and existing components and devices are being continuously improved. As in digital computing, the need for higher speeds of operation is bringing forth some interesting ideas. One example is a new high-speed correlation computer developed for automatic data processing. This shortens considerably the time required for analysing a signal recorded on a length of magnetic tape by automatically scanning the tape in sections.

In orthodox correlation computers, as has been explained in Chapter 6, the entire length of tape has to be run through for each spacing of the two playback heads (giving correlation values at different time intervals). In the new high-speed machine, however, each section of tape to be analysed is held stationary on a drum while the playback heads rotate about it. At the same time the heads are automatically displaced in steps to produce points in the correlation function graph. Both auto-correlation and cross-correlation analyses can be performed on the machine.

In d.c. analogue computers of the repetitive kind, as described in Chapters 3, 4 and 5, one way of speeding up the computation is simply to increase the repetition rate. One experimental machine, at King's College, University of London, operates at a rate as high as 25,000 per second. The really interesting feature of this computer, however, is that it enables a whole series of solutions with different constant-multiplier settings to be obtained automatically, and at such high speed that they can be presented simultaneously on the screen of a cathode-ray tube. The constant-multiplier resistors are varied in steps—not, of course, by manually operated switches but by electronic switching circuits. As a result the machine does not produce a single trace like that shown in Fig. 5.3, but a whole bank of them, each slightly different from the previous one as the constants are changed. These traces are arranged to build up a projective three-dimensional display on the screen rather analogous to a solid contour map or model. Thus the computer not only calculates at high speed (a hundred to a thousand times faster than conventional analogue computers) but presents the results in a form which allows a large volume of information to be assessed quickly and conveniently.

Another improvement in the operation of analogue computers is the use of digital systems for input, output and control purposes. As an example a " digital output/input translator " system is used on a Beckman Instruments analogue machine to automatically set up and check problems by means of punched tape and electric typewriters. The computer can be set for initial conditions, computing or checking by the programme; coefficient potentiometers can be automatically set or read and their values printed out; and various points in the computer circuit can be scanned and their voltages and signs be printed out. The punched tape can be used as a means of storing information on setting values, so that the computer can be cleared for other work and then the original information can be reinserted later.

One of the main disadvantages of the analogue computer, compared with the digital machine, is its limited accuracy. A type of equipment which combines the speed and flexibility of the analogue machine with the accuracy of the digital computer is the digital differential analyser. This, as its name implies, is digital in operation but is organized to have the general functional properties of the analogue differential analyser (see Chapter 4).

The Royal Aircraft Establishment have designed one machine of this kind for accurate simulation work on control systems. The basic element is a device which performs approximate integration by finite differences—roughly analogous to the principle of totalling squares on graph paper underneath a curve. Several such integrators are required for solving differential equations and each integrator has storage registers associated with it for holding the running totals of increments which occur in the process. There is, however, only a single computing circuit which is used in turn by the integrators, as it would be too expensive to provide one for each. Because all the quantities are handled as pulse counts, the digital differential analyser has the advantage that it can generate functions of any variable and not merely functions of time as in the time-dependent analogue machine.

14

COMPUTERS OF THE FUTURE

THE economic survival of a small industrial country like the United Kingdom will depend very much on the successful exploitation of automatic methods of production. It is in this sphere that computers of the future are likely to play their most important role. For the moment automation in industry is concerned with the automatic operation of machines and processes, while leaving the control and organization of the factories in the hands of human beings. The ultimate aim of automation, however, is much more ambitious than this. As we have already hinted in Chapter 12, the idea is to use computers as central " brains " or " data clearing houses " for the overall control and co-ordination of complete factories.

Hitherto computers have been isolated in laboratories and business offices and used merely as aids to human calculation. Even when employed for production planning in factories, they have needed human supervisors to feed in and take out information. But there are signs of a change in this situation. Already computers are beginning to appear as components of complex automatic control systems in which many variables have to be taken into account. What this trend of development may lead to in the future has been imaginatively forecast by Norbert Wiener, the American mathematician, in his book, " The Human Use of Human Beings ".* Describing a hypothetical automobile factory, he says: " In the first place, the sequence of operations will be controlled by something like a modern high-speed computing machine. In this book and elsewhere I have often said that the high-speed computing machine is primarily a logical machine, which confronts different propositions with one another and draws some of their consequences . . .

---

* Eyre & Spottiswoode, revised edition, 1954.

" The instructions to such a machine, and here, too, I am speaking of present practice, are given by what we have called a taping. The orders given the machine may be fed into it by a taping which is completely predetermined. It is also possible that the actual contingencies met in the performance of the machine may be handed over as a basis of further regulation to a new control tape constructed by the machine itself, or to a modification of the old one . . .

" The computing machine represents the centre of the automatic factory, but it will never be the whole factory. On the one hand, it receives its detailed instructions from elements of the nature of sense organs, such as photo-electric cells, condensers for the reading of the thickness of a web of paper, thermometers, hydrogen-ion-concentration meters, and the general run of apparatus now built by instrument companies for the manual control of industrial processes. These instruments are already built to report electrically at remote stations. All they need to enable them to introduce their information into an automatic high-speed computer is a reading apparatus which will translate position or scale into a pattern of consecutive digits. Such apparatus already exists, and offers no great difficulty, either of principle or of constructional detail. The sense-organ problem is not new, and it is already effectively solved.

" Besides these sense organs, the control system must contain effectors, or components which act on the outer world. Some of these are of a type already familiar, such as valve-turning motors, electric clutches and the like. Some of them will have to be invented, to duplicate more nearly the functions of the human hand as supplemented by the human eye . . .

" Of course, we assume that the instruments which act as sense organs record not only the original state of the work but also the result of all the previous processes. Thus the machine may carry out feedback operations, either those of the simple type now so well understood, or those involving more complicated processes of discrimination, regulated by the central control as a logical or mathematical system.

" In other words, the all-over system will correspond to the complete animal with sense organs, effectors and proprioceptors, and not, as in the ultra-rapid computing machine, to an isolated brain, dependent for its experiences and for its effectiveness on our intervention . . . ."

Undoubtedly Wiener's description is very much of a pipe-dream at the moment (or should we say a pipe-nightmare?), but it comes from an informed imagination and should not be dismissed too lightly. Already, in fact, we are seeing the initial developments. The use of a computer as part of a process control system is a practical reality (see Chapter 12) and soon it will become commonplace.

The most straightforward application in industry is probably where the process has a continuous nature—as, for example, in a chemical plant. The monitoring, calculation and error-correction then proceeds in a more or less continuous manner rather on the principle of a servomechanism. In a chemical plant the object would be to keep the plant output as near as possible to a desired specification. The outputs from local control loops already existing in the plant would be continuously monitored in a scanning sequence and compared in the computer with the desired values for the product. According to the results, the computer output would be used to apply correcting signals to the control-loop equipment.

For this kind of application the computer would probably be a parallel type in order to attain high speed for " real time " operation—though in certain industrial plants the " real time " of the process might be quite slow and high-speed computing would not be necessary. The machine would have a multiplicity of input and output units distributed about the factory, and it is possible that the computing circuits themselves would also be decentralized and distributed. Electronic components such as transistors, magnetic cores and cryotrons would probably be used in these circuits so that voltages could be kept to low values for safety reasons.

A rather different type of control situation would occur where the process was not continuous—where the product consisted of a number of separate items, like motor cars, each with a particular identity. As an example, there might be a conveyer belt system in which the products on the belt could be varied in design (e.g., different paint colours on motor car bodies) according to the day-to-day fluctuations in demand from customers. In the control apparatus the movement of the conveyer belt might be simulated by a storage medium moving at a corresponding speed past a complexity of reading and writing heads connected to the digital computer. Numerically coded information about the items on the belt would be written by the computer at corresponding places on the storage medium. As a particular item reached a " decision point " on the conveyer belt, the stored information on what was to be done at that point would come to a reading head, which would give out the appropriate signals for actuating the process.

All these ideas, of course, are based on our knowledge of what computers are capable of doing in their present form.

Computers are, however, developing all the time, and one is entitled to wonder if their abilities are likely to be extended in the future. Will they, for example, be able to handle problems of a more subtle and complex nature than are envisaged at the moment?

Undoubtedly the computer is wonderful enough already, but its virtues are those of the slave and the drudge. It will get through phenomenal amounts of work at tremendous speed, but every single step of its operations has to be worked out beforehand in great detail. It will do exactly what it has been asked to do but nothing more. The analogue computer has to have its various elements connected into a suitable circuit while the digital machine has to be programmed. From certain types of mechanisms, of course, one does not expect any more. Nevertheless, in this modern world of servo-mechanisms and self-regulating systems one feels very much inclined to ask why the computer cannot be given a greater degree of automaticity than it has at present. Why should we not make it less like an uncomprehending slave and more like a skilled craftsman, with some degree of free will and ability to adjust itself to changing circumstances? The answer is, in many cases, that we prefer to keep the computer slightly stupid—that it would just not do for a machine to go making up its own mind about things. In the future, however (and we are talking about the future), there may well be some applications, particularly in industrial control systems, where this sort of ability could be extremely useful. For example, it might be desirable for a computer in a process control system to be capable of dealing with unforeseen circumstances—behaviour outside the scope of normal programming. Here, a degree of adaptability approaching that of human intelligence would be called for. Then there are some industrial processes which do not lend themselves to precise mathematical analysis, and here again it might not be possible to use the conventional type of computer for control purposes.

Scientific workers in universities and research organizations are already coming to grips with this problem of " intelligent automation " and a variety of experimental machines have been built to try out theories. In some cases, where it has been too expensive to construct special machines, simulation of their properties has been carried out on existing computers (see Chapter 12). In other cases computers have been programmed to simulate intelligent human behaviour merely as academic exercises—to discover what is technically feasible.

In the digital computer, the conditional operations mentioned in previous chapters offer some interesting possibilities, for they endow the machine with the power of discrimination and apparent decision. This ability is not peculiar to computers, of course, and has been built into much simpler electronic devices, but it can produce some rather lifelike behaviour when used in conjunction with other facilities available in a digital machine.

For example, suppose that the programme sequence is arranged to proceed in either of two channels, depending on whether a number in a particular trigger store is above or below a certain value. What this number actually is can be made to depend, in turn, on the magnitude of a number fed into the computer from outside, so that the course of action taken by the machine is decided by what might be called the strength of an external stimulus.

The course of action which is initiated in this way can be arranged to modify the number in the trigger store, perhaps increasing it for one course of action and decreasing it for the other.

If then a second " stimulus " number is applied to the computer, one discovers that the response of the machine does not depend exclusively on the strength of this second stimulus, but partly on the first one as well. Moreover, as a result of the second stimulus, there is another modification of the number in the trigger store, and this again alters the conditions so that the machine's response to a third stimulus will be dependent on the first two. This process of modification goes on and on with more and more stimuli, so that the response of the computer to any particular stimulus is always influenced by the whole history of the previous ones and the operations resulting from them.

One might say, in fact, that the computer's behaviour becomes controlled by its previous experiences, which can be considered as a very elementary form of learning. It is possible to take the principle even further and achieve an imitation of the biological conditioned reflex phenomenon, which is also regarded as a basis for learning. This has been done on the EDSAC computer at Cambridge by A. G. Oettinger. The aim of the experiment was to obtain a certain desired response from the machine (say a particular number in the output) when it was activated by a stimulus, by applying other signals representing either approval or disapproval. At first the computer tended to give random responses, but after a period in which the " approval " signal was applied for every correct response and the " disapproval " one for every incorrect response, it " learnt " to give the desired response every time.

If, after this, the stimulation was continued without the " approval " or " disapproval " signals, the machine gradually forgot what it had learnt and began to give unwanted figures at the output.

Analogue computers also have been used to imitate this process of learning by association. They usually take the form of simulators, specially constructed by neurophysiologists as " models " of nerve networks. The behaviour of interacting groups of nerve cells can actually be described by mathematical equations which could, in principle, be solved on a digital computer. The process would take far too long, however, and even then it would give only one solution at a time. As in aircraft design, it is much more useful to have a computer which gives a continuous series of different solutions and acts to some extent as a model of the real system—and this is where the simulator comes in.

A particularly interesting analogue machine of this type has been built by W. K. Taylor at University College, London. It is made up of model nerve cells, each containing seven valves, which can be adjusted to simulate the properties of living cells and be interconnected in various ways to represent different nerve networks. Some of the model cells can be used as input units (equivalent to the nerve cells in sense organs) and others as output units (corresponding to the nerves controlling muscle activity) so that stimuli can be applied, and the responses to them observed, in the same way as with living organisms.

In the process of learning by association the device is regarded as a kind of animal. Two types of stimuli, equivalent to " pain " and " food ", are applied to suitable input units and the machine produces appropriate responses of " escape " and " advance " at the output units. At the same time two types of visual patterns can be applied to other input units, and what the machine learns to do is to associate these with the stimuli so that one pattern comes to mean " pain " and the other " food ". At first the machine is not successful, but after a number of trials the two separate associations are formed, and then if the patterns are presented alone the correct responses to them are obtained without the original stimuli.

A rather simpler type of analogue machine which learns by association has been devised by W. Grey Walter, of the Burden Neurological Institute, Bristol. It is really intended to imitate the conditioned reflex experiments conducted by the Russian scientist Pavlov. (In these a dog was conditioned to associate the sound of a bell with food, by presenting them together several times, after which it responded to the bell alone as if that were food.) Regarded as a computer, the analogue device really works on statistical principles. It calculates the probability of the two input signals (" bell " and " food ") being related, and if this is above a certain value the information is stored and used to open a gate circuit which allows the " bell " signal to produce the same response as the " food " signal. Incidentally, the device has been incorporated in one of the mobile electro-mechanical robots or " tortoises " for which Grey Walter has become well known (see p. 231, Plate 31 [c]).

Devices of this kind are, of course, specifically designed to do the job of imitating physiological mechanisms. At the same time it may be interesting to note that certain neurophysiologists have compared the animal brain and nervous system with existing computers, particularly digital machines. This analogy has arisen from the fact that the basic units of the nervous system—the neurons, or nerve cells—can be considered as two-state devices like those which perform binary operations in a digital computer: they are either conducting or non-conducting. Moreover, a great many neurons act in a similar way to the coincidence-detector type of gate circuits described in Chapter 7: they give an output pulse only when triggered by a certain pattern of pulses (from other cells) at the input. The nerve cell is actually triggered by a time-sequential pattern, as distinct from the simultaneous pattern applied to the electronic circuits, but the logical outcome is the same.

Comparisons of this kind are not very popular at the moment amongst the electro-physiologists, but two other points of resemblance are worth mentioning. The first is to do with the " memory " or storage of information. As explained in Chapters 9 and 10, one method of storing number pulses in the digital computer is to keep them circulating in some form of delay-line, regenerating them every time round to preserve their amplitude and shape. The fact that storage can be achieved in this way has prompted the neurophysiologists to think in similar terms about the brain, and they have suggested that at least a part of our memory system consists of long and tortuous loops of nerve cells in which signals are kept circulating by regenerative action.

The other point of resemblance is concerned with the " clock " in the digital computer (see Chapter 8), which determines the actual instants at which the various operations occur. Drawing an analogy from this, neurophysiologists have suggested that in the brain a similar timing function may be performed by the 10 c/s electrical waveform known to electro-encephalographers as the "alpha rhythm". At any rate, experiments on human subjects have shown that certain control signals in the body (affecting muscle activity) appear to be " locked " in a definite phase relationship with this 10 c/s waveform.

To return to " learning " operations, another method of learning is by the trial-and-error process, and this also has been imitated in a number of computing machines. It is possible, for example, to programme a digital computer to search a number of storage locations to find a particular item of information. At first the machine searches in a random fashion, but once the information has been found it registers the location in its memory system and on the next occasion goes straight to it without hesitation. There are also devices known as maze-solvers which are specially designed to do the same sort of thing. In these a " mechanical mouse " blunders about in a maze until it finds its way to a goal by trial and error. Put in the maze a second time the " mouse" goes straight to the goal without mistakes, being controlled by a memory system which has recorded only the successful moves from the previous run.

For industrial control purposes, the mode of operation of the " learning " process would be to try tentatively various patterns of control action in a random fashion and to reinforce those which gave satisfactory results and inhibit those which gave unsatisfactory results. As already explained it is possible for existing computers to perform this type of operation.

In practice, however, computers would not be used because they are not designed for such applications—their normal facilities would be largely wasted and they would be needlessly expensive. Special machines would be designed for the purpose, and, in fact, experimental systems are already being tried out in many laboratories.

An outline illustration of how the trial-and-error learning process would work in an industrial plant has been given by A. M. Andrew* of the National Physical Laboratory. The block diagram of Fig. 14.1 shows the essentials of a continuous-process industrial control system, with raw materials flowing in on the left and the final product emerging on the right. Variables such as temperatures, pressures and flow-rates are regularly monitored and the information is used by the "controller" to make suitable corrections to the process. The transducer on the right measures to what extent the final product has the desired properties (which might be, say, quality or quantity).

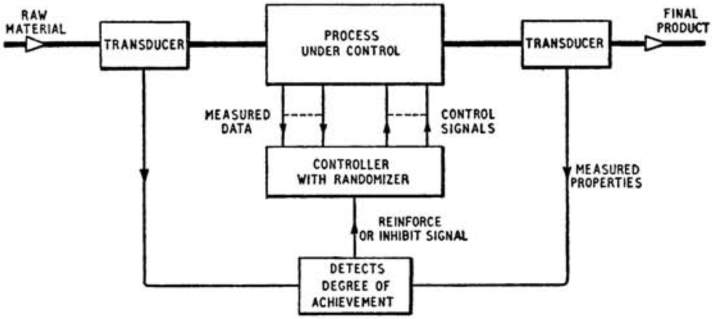


Fig. 14.1. Continuous-process industrial control system with properties of trial-and-error learning

So far this is quite conventional, but the "controller" also contains a "randomizer", the purpose of which is to try different methods of interpreting the measured information. More precisely it varies the values of the mathematical functions by which the controlling signals are related to the monitoring signals.

---

* In a paper "Learning Machines" presented at a Symposium on the Mechanization of Thought Processes at the N.P.L., November, 1958.

When the final product tends towards the desired properties as a result of a fortuitous trial by the " randomizer ", the transducer on the right causes a signal to be sent to the " controller " which reinforces the control action leading to this result. If the product tends in the other direction because of an unlucky random trial, an inhibiting signal is sent to reduce the control action giving rise to the unwanted tendency. This " approval " or " disapproval ", as it were, of the randomly selected control patterns may also be made partly a function of the raw material properties, as indicated by the transducer on the left.

Thus the learning system adjusts itself, not, as in a conventional control system, by using predetermined functions of the measured physical variables, but by using any convenient functions which lead to optimum conditions in the final product.

The terms " reinforce " and " inhibit " used in the foregoing are just convenient words to describe the processes which would achieve these results. Dr. Andrew suggests several different methods of instrumentation. Some are based on a special-purpose conditional probability computer —a machine which computes on a statistical basis the probabilities that certain events are related. The method here is to discover on a statistical basis what patterns of control action (applied to some of the computer inputs) are associated with " success " signals from the final-product transducer (applied to another input). This type of computer, however, works on a two-state " yes-no " principle and is not really suitable for continuously varying quantities. A more straightforward method of association is based on correlation computing (see Chapter 6). Here the idea is to make running correlations between the degree of goal-achievement in the final product and the values of the random mathematical functions tentatively tried out in the " controller " of the block diagram. If there is a positive correlation between the variation of one of these values and the variation in the final goal-achievement, then the learning system increases the value concerned (i.e., reinforcement). Conversely if there is negative correlation the learning machine decreases the value (inhibition).

The conditional probability computer mentioned earlier has been demonstrated, purely for illustrative purposes, as the " brain " of a mechanical " tortoise " which follows a complicated boundary between black and white areas. Here the learning system forms associations between the " black " or " white " signals the tortoise picks up from its photo-electric receptors in front and the corresponding signals indicating " right " or " left " movements which it makes to regain the boundary. As in the industrial process control system, the correct associations are formed by applying reinforcing or inhibiting signals to the control patterns according to the degree of goal-achievement (in this case, regaining the boundary). Thus the tortoise can learn different patterns of association between its sensory perceptions and its motor activity. As a result it can cope with situations such as reversal of the positions of the black and white areas or reversal of the steering linkage.

Similar flexibility of action is achieved by this type of learning machine in another sphere of " intelligent " automation—the recognition of patterns. This will have practical application in devices for translating, say, handwritten characters into speech or print, speech into typescript, or similar visual or acoustic stimuli into mechanical activity. The great problem in pattern recognition is to be able to imitate the human faculty of recognizing patterns even if badly distorted or placed in unfamiliar positions or circumstances. Most people can recognize ∀ in a normal line of type as being an A upside-down, but a character-recognizing machine working on normal principles (see Chapter 13) would, with all due respect, be completely lost. Not so a " learning " character recognizer of the kind developed by W. K. Taylor on the basis of his nerve analogue machine (see p. 251). Here the general principle is that the machine learns to classify various distorted or unfamiliar versions of the patterns with the normal forms, so that on the occasions when a distorted or unfamiliar version is presented to the recognizer it still gives the correct output signal.

The machine uses a matrix of photomultipliers as an " eye " to analyse the pattern presented into small black and white areas. The learning mechanism itself is based on an assemblage of electronic units which are analogues of nerve cells, as mentioned earlier. These are connected together to permit an action corresponding to the " facilitation of synapses " which is thought to be a possible basis for learning in animals and humans. It simply means that certain junctions between nerve cells have their ability to transmit nerve impulses increased by the process of learning, and the pattern of these " facilitated " transmission paths is a stored representation of what is learned—a memory, in fact.

One can see, then, from the general picture which emerges of these " intelligent " machines, that some of the computers of the future will not work on the established principles which have been the main subject of this book. They will not be " deterministic " in operation (to borrow a term from philosophy) but " probabilistic ". In other words, they will be concerned with computing probabilities on a statistical basis, rather as human beings appear to do when making intelligent guesses. This probabilistic method of thinking is perhaps more fundamental to human behaviour than the logical processes of deduction and calculation which are imitated by conventional computers.

Another distinguishing feature of the " intelligent " machines is that their internal connections are not fixed by the design or predetermined by a human programmer, as in conventional computers. They depend either on chance or on the results of past experience, and even the machine designer cannot predict what pattern they will take up at any given moment. It is an intriguing thought that machines can thus be endowed with some degree of " free will ".

These two attributes, of probabilistic calculation and adaptation to suit the circumstances, have so far only been obtainable from human beings for the control of industrial plant and other complex systems. Now that they can be obtained from machines as well we are certainly on the way to the new era of " intelligent " automation.

INDEX

Abacus, 14
Accumulator (accumulating register), 107
Accuracy, analogue *vs.* digital, 15
ACE digital computer, 11
Adding,
  analogue, 35
  binary digital, 99, 104
Addresses,
  in programming, 186
  of numbers, 123
Aerodynamic calculations, 79
Aiken, Howard, 9
Aircraft control, automatic, 216
Analogue,
  computing principle, 14
  –digital comparison, 15, 22
Analytical engine, Babbage's, 5
AND gate, 100
Automatic character recognition, 235
Automatic programming, 194
Automobile design calculations, 88

Babbage, Charles, 5
Backing-up stores, 133, 161
Binary,
  addition rules, 99
  coded decimal computing, 32, 110
  –decimal equivalents, 31
  number representation, 27, 97
Bi-stable transistor circuit, 220
Boole, George, 6
Brain-computer comparisons, 253
Bush, V., 8

Cards, Jacquard, 7
" Carry " operation in digital computers, 4, 22, 101
Cathode-ray tube store, 150
Character recognizers, 235, 257
Clock pulses, 119
Coded instructions for programming, 186
Coincident-current magnetic matrix store, 138
Conditional operation in digital computers, 23, 129, 190, 193, 250
Conditional probability computer, 256
Conditioned reflex imitation, 251, 252
Control circuits, digital computer, 115
Control desk, digital computer, 181
Correlation computer, 91
Correlation computing in learning mechanisms, 256
Counting, mechanization of, 4, 10, 22
Crossed-fields multiplier, analogue, 46
Cryogenic devices, 226

D.C. amplifier, analogue, 38, 58
Data processing, 197
Decimal–binary conversion, 31, 191
Decimal counter tubes, 32
Decoders, in digital computers, 125
Delay-line storage, 118, 147
Delay-lines, electrical, 149
Differential analyser,
  analogue, 25, 52
  digital, 243
Differentiation, analogue, 44
Digital–analogue,
  comparison, 15, 22
  computer size classification, 157
  differential analyser, 243
  simulators, 214
Direct-coupled transistor circuits, 220

Drift correction in d.c. amplifiers, 59, 61, 62
Drum, magnetic, storage, 144, 162

Eccles-Jordan trigger circuit, 10, 116
EDSAC computer, 11
EDVAC computer, 11
Electrostatic storage, 150
ENIAC computer, 10

Factory control by computer, 245
Ferroelectric storage, 139, 234
Ferroresonant store, 140
File computers, 209, 210
Fixed-programme digital computers, 159
Flight simulation, 79
Floating-point operation in digital computing, 194
Flow diagram for programming, 185
Four-layer semiconductor diode, 226
Function generators, analogue, 48

Gates, electronic, 99
General-purpose analogue computers, 57

Half-adder, digital, 103
Helical magnetization stores, 230
High-frequency transistors, 219
Hollerith, 7

Indicators for analogue computers, 70
Initial conditions in analogue computers, 71
Input/output equipment,
    magnetic-tape, 163
    punched cards, 165
    punched paper tape, 169
Instruction codes, programme, 186
Integration with respect to time, 25, 41
" Intelligent " machines, 249

Jacquard, 7
Jevons, W. S., 6
" Jump " operation in digital computers, 129, 190, 193

Kelvin, Lord, 7, 8

Language translation by computer, 209
" Learning " processes in computers and related devices, 251, 254
Leibnitz, 5
LEO computer, 2
  applications, 200
Line printers for computer output, 168
Linear programming, 208

Machine-tool control,
    analogue, 94
    digital, 211
Magnetic-core,
    logic circuits, 221
    storage, 137
    transistor logic circuits, 223
Magnetic-drum storage, 144, 162, 164
Magnetic-film stores, 233
Magnetic-tape,
    input/output equipment, 163, 179
    storage, 141, 163
Magnetostriction delay-line store, 147
Main store of digital computer, 123
Majority-decision logic, 221
Matrix magnetic-core store, 137
Mental calculation processes, 16
Mercury delay-line stores, 148
Military data processing, 212
Multiplication,
  analogue, 45
  binary digital, 108
Multiplying by constant factor, analogue, 36

Napier, John, 4
Negative feedback in d.c. amplifiers, 37
Negative-resistance two-state devices, 225
Nerve network analogue, 251
Non-numerical data processing, 209
Non-return-to-zero magnetic storage, 143
Nuclear reactor simulator, 90

INDEX

Number,
  magnitude in digital computers, 194
  period in digital computers, 119

Office work mechanization, 198
Optimum programming, 131
OR gate, 99

Packaged units, 158
Paper tape for digital input/output, 169
Parallel operation in digital computers, 113
Parametrons, 234
Pascal, Blaise, 4
Patch panel, analogue computer, 68
Payroll preparation using digital computers, 200
Plug-board programming for digital computers, 159
Predictors, 8, 14
Printed storage systems, 229
Probabilistic calculation, 258
Process control using digital computers, 198, 215, 247
Production planning using digital computers, 202
Programming,
  codes, 186
  digital computers, 23, 115, 183
  flow diagram, 185
Punched cards,
  for input/output, 165
  reading, 167
Punched paper tape for input/output, 169

Quick-access storage, 133

Random-access storage, 146
"Real time" operation, 76, 113, 215, 247
Register, storage, 106
Repetitive operation in analogue computers, 62
Return-to-zero magnetic storage, 143

Scientific applications of digital computers, 213
Sequence control, 9, 10, 23, 114
Serial operation in digital computers, 113, 115
Servo-multiplier, analogue, 45
Sign reversal, analogue, 41
Simulators, 9, 76, 214, 251
Single-address digital computer, 131
Size of digital computers, 157
Solid circuitry, 225
Speed of digital input/output equipment, 180
Square-loop magnetic material for storage, 137
Square-root sub-routine, 193
Static storage of binary numbers, 97, 116, 134
Statistical computer, 95
Step function input for analogue computers, 72
Storage system requirements, 132
Sub-routine for decimal-binary conversion, 167
Sub-routines, 23
  in programming, 192
Subtraction,
  analogue, 36
  binary digital, 107
Superconductive computing elements, 226

Tape, magnetic storage, 141, 163
Time sharing in process control applications, 215
Transfer functions, 69, 77
Transistor,
  advantages, 218
  diode, 226
  /magnetic-core logic circuits, 222
  OR gate, 221
  trigger circuit, 220
Translation,
  of languages by computer, 209
  programme for automatic coding, 195
Trigger circuit, Eccles-Jordan, 10, 116
Two-input adder, digital, 103

Voltage analogue of numbers, 24, 35

Xerographic output printer, 240

Indispensable
to technicians

Britain's leading technical journal covering radio, television and electronics, WIRELESS WORLD provides detailed surveys of the newest techniques in design and manufacture. Articles of a high standard cover every phase of radio and electronic practice.

Wireless World

Monthly 2s. Annual subscription £1 15s. 0d.

Month by month, ELECTRONIC TECHNOLOGY covers world progress in electronics, radio and television. Advanced articles on theory, design and development include many original papers by physicists and engineers of international repute. Its important Abstracts and References Section forms one of the finest sources of technical and research information available.

**Electronic
technology**

Monthly 5s. Annual subscription £3 7s. 0d.

From all newsagents or direct :—
ILIFFE & SONS LTD.
DORSET HOUSE, STAMFORD ST., LONDON, S.E.1

ABACS OR NOMOGRAMS

A. Giet. Translated from the French by H. D. Phippen and J. W. Head. Demonstrates the many and varied applications of the Abac or Nomogram and shows how even those without highly specialised mathematical knowledge may construct their own charts. Specially adapted for English readers, this edition will prove of the greatest value to engineers and physicists, and many others who require time-saving methods for repetitive and complicated calculations.

8¾″ × 5½″. 235 *pp.* 152 *illus.* 35*s. net. By post* 36*s.*

BASIC MATHEMATICS FOR RADIO AND ELECTRONICS

F. M. Colebrook, B.Sc., D.I.C., A.C.G.I. Revised by J. W. Head, M.A.(Cantab.), Mathematical Consultant, Research Department, B B C Engineering Division. Third Edition. Presents, in a very readable form, a complete course in fundamental mathematics for engineering students. Also covers the application of mathematics to radio, Stability, Linear Differential Equations, Elementary Statistics, Short Cuts to Numerical Calculations and an Introduction to Matrices.

7½″ × 5″. 359 pp. 17*s.* 6*d. net. By post* 18*s.* 6*d.*

LAPLACE TRANSFORMS FOR ELECTRICAL ENGINEERS

B. J. Starkey, Dipl.Ing., A.M.I.E.E. A presentation of the theory of the Laplace transformation in which a physical vocabulary rather than a purely mathematical one is used as far as possible. This method of analysis has become of increasing importance to electrical engineers in many fields during recent years, and the work is designed to provide a thorough treatment of the subject in a language with which they will be familiar.

8¾″ × 5½″. 279 *pp.* 30*s. net. By post* 31*s.* 2*d.*

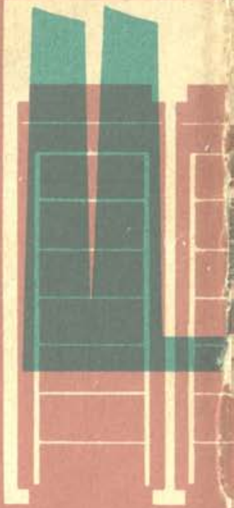
PRINCIPLES OF TRANSISTOR CIRCUITS

S. W. Amos, B.Sc.(Hons.), A.M.I.E.E. An introduction to the design of transistorised equipment for amateur designers, students and amateur constructors. Deals with the physical processes occurring in transistors, the application of these principles to practical problems of design, and the determination of such quantities as input resistance, stage gain, optimum load, power output, values of coupling capacitors and transformer winding inductances.

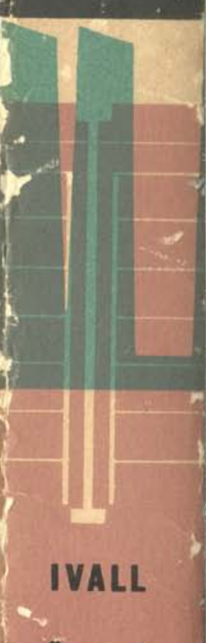
8¾″ × 5½″. 167 *pp.* 105 *diagrams.* 21*s. net. By post* 21*s.* 11*d.*

Obtainable from leading booksellers

ILIFFE & SONS LTD., Dorset House, Stamford Street, London, S.E.1



ELECTRONIC COMPUTERS



IVALL

Second
Edition

ILIFFE